Hao (Jeff) Deng ID:932912420

For this homework assignment, I will be using **LINDO** to calculate the problems.

1). A). Find the distance of the shortest path from G to C in the graph below.

Objective function is max c as to find the shortest path to c from g, and the constraints are the code between ST and END below from the screenshot in which they are the distances from one vertex to another. For example, h-g <= 3 is interpreted as the distance from h to g is 3.

The shortest path from g to c is 16.

Copy of Code

```
max c
ST
    g=0
    h-g<=3
    d-g<=2
    g-e<=7
    e-d<=25
    d-e<=9
    e-b<=10
    b-h<=9
    a-h<=4
    e-f<=2
    b-a<=8
    a-f<=5
    f-a<=10
    b-f<=7
    c-b<=4
    c-f<=3
    f-d<=18
    d-c<=3
END
```

Result (Output)

LP OPTIMUM FOUND AT STEP | 6

OBJECTIVE FUNCTION VALUE

1) 16.00000

VARIABLE	VALUE	REDUCED COST
C	16.000000	0.000000
G	0.000000	0.000000
H	3.000000	0.000000
D	0.000000	0.000000
E	0.000000	0.000000
B	12.000000	0.000000
A	4.000000	0.000000
F	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)	7.000000	0.000000
6)	25.000000	0.000000
7)	9.000000	0.000000
8)	22.000000	0.000000
9)	0.000000	1.000000
10)	3.000000	0.000000
11)	15.000000	0.000000
12)	0.000000	0.000000
13)	14.000000	0.000000
14)	1.000000	0.000000
15)	8.000000	0.000000
16)	0.000000	1.000000
17)	0.000000	0.000000
18)	5.000000	0.000000
19)	19.000000	0.000000

NO. ITERATIONS= 6

LP Code

```
max c
ST
        g=0
        h-g<=3
        d-g<=2
        g-e<=7
        e-d<=25
        d-e<=9
        e-b<=10
        b-h<=9
        a-h<=4
        e-f<=2
        b-a<=8
        a-f<=5
        f-a<=10
        b-f<=7
        c-b<=4
        c-f<=3
        f-d<=18
        d-c<=3

END
```

B). Find the distances of the shortest paths from G to all other vertices.

The objective function is max a+b+c+d+e+f+h to find the shortest paths from g to all the vertices. Although we will get the sum of all the shortest paths, but that doesn't matter because we only care about the shortest path in each vertex from g, which is underneath the objective function value in the picture below. The constraints the code between ST and END below from the screenshot in which they are the distances from one vertex to another. For example, h-g <= 3 is interpreted as the distance from h to g is 3.

The shortest path from G to all the other vertices:

A	B	C	D	E	F	H
7	12	16	2	19	17	3

The value "76" is meaningless because that is the sum of all the shortest paths, but the shortest paths have been listed below from A to H.

Copy of Code

```
max a+b+c+d+e+f+h
ST
        g=0
        h-g<=3
        d-g<=2
        g-e<=7
        e-d<=25
        d-e<=9
        e-b<=10
        b-h<=9
        a-h<=4
        e-f<=2
        b-a<=8
        a-f<=5
        f-a<=10
        b-f<=7
        c-b<=4
        c-f<=3
        f-d<=18
        d-c<=3
END
```

Result (Output)

```
LP OPTIMUM FOUND AT STEP      4

       OBJECTIVE FUNCTION VALUE

       1)      76.00000

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

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

NO. ITERATIONS=       4
```

LP Code

```
max a+b+c+d+e+f+h
ST
        g=0
        h-g<=3
        d-g<=2
        g-e<=7
        e-d<=25
        d-e<=9
        e-b<=10
        b-h<=9
        a-h<=4
        e-f<=2
        b-a<=8
        a-f<=5
        f-a<=10
        b-f<=7
        c-b<=4
        c-f<=3
        f-d<=18
        d-c<=3

END
```

2). Formulate the problem as a linear program with an objective function and all constraints. Determine the optimal solution for the linear program using any software you want. Include a copy of the code and output. What are the optimal numbers of ties of each type to maximize profit?

Type	Selling Price	Labor Cost	Material Cost	Profit
Silk	6.7	0.75	2.5	3.45
Polyester	3.55	0.75	0.48	2.32
Blend 1	4.31	0.75	0.75	2.81
Blend 2	4.81	0.75	0.81	3.25

Formulate the problem as linear program

max 3.45s + 2.32p + 2.81b + 3.25c
ST

0.125s <= 1000 : This is for Silk
0.08p + 0.05b + 0.03c <= 2000 : This is for Polyester
0.05b + 0.07c <= 1250 : This is for Cotton
s <= 7000; s >= 6000
p <= 14000; p >= 10000
b <= 16000; b >= 13000
c <= 8500; c >= 6000

Copy of Code

```
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 <= 7000
    s >= 6000
    p <= 14000
    p >= 10000
    b <= 16000
    b >= 13000
    c <= 8500
    c >= 6000
END
```

Result (output)

LP Code

```
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 <= 7000
        s >= 6000
        p <= 14000
        p >= 10000
        b <= 16000
        b >= 13000
        c <= 8500
        c >= 6000

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)         0.000000          3.450000
       6)      1000.000000          0.000000
       7)       375.000000          0.000000
       8)      3625.000000          0.000000
       9)      2900.000000          0.000000
      10)       100.000000          0.000000
      11)         0.000000          0.476000
      12)      2500.000000          0.000000

NO. ITERATIONS=       4
```

The maximum profit is $120,196.00 from producing 7,000 units of silk ties, 13,625 units of polyester ties, 13,100 units of Blend 1, and 8,500 units of Blend 2.

3). A). Determine the number of refrigerators to be shipped from the plants to the warehouses and then warehouses to retailers to minimize the cost. Formulate the problem as a linear program with an objective function and all constraints. Determine the optimal solution for the linear program using any software you want. What are the optimal shipping routes and minimum cost?

Objective Function:

Minimize transshipment costs (Z) = 10P1W1 + 15P1W2 + 11P2W1 + 8P2W2 + 13P3W1 + 8P3W2 + 9P3W3 + 14P4W2 + 8P4W3 + 5W1R1 + 6W1R2 + 7W1R3 + 10W1R4 + 12W2R3 + 8W2R4 + 10W2R5 + 14W2R6 + 14W3R4 + 12W3R5 + 12W3R6 + 6W3R7

Constraints:

Constraints on Plants (P):

P1W1 + P1W2 <= 150
P2W1 + P2W2 <= 450
P3W1 + P3W2 + P3W3 <= 250
P4W2 + P4W3 <= 150

Constraints on Retailers (R):

W1R1 >= 100
W1R2 >= 150
W1R3 + W2R3 >= 100
W1R4 + W2R4 + W3R4 >= 200
W2R5 + W3R5 >= 200
W2R6 + W3R6 >= 150
W3R7 >= 100

Constraints on Warehouses (W) since nothing should be stored in the end

W1R1 + W1R2 + W1R3 + W1R4 – P1W1 – P2W1 – P3W1 = 0
W2R3 + W2R4 + W2R5 + W2R6 – P1W2 – P2W2 – P3W2 – P4W2 = 0
W3R4 + W3R5 + W3R6 + W3R7 – P3W3 – P4W3 = 0

Non-negativity Constraints:

P1W1 >= 0
P1W2 >= 0
P2W1 >= 0
P2W2 >= 0
P3W1 >= 0
P3W2 >= 0
P3W3 >= 0
P4W2 >= 0
P4W3 >= 0

W1R1 >= 0
W1R2 >= 0
W1R3 >= 0
W1R4 >= 0
W2R3 >= 0
W2R4 >= 0
W2R5 >= 0
W2R6 >= 0
W3R4 >= 0

W3R5 >= 0
W3R6 >= 0
W3R7 >= 0

Hao (Jeff) Deng ID:932912420

LP Code

```
min 10P1W1 + 15P1W2 + 11P2W1 + 8P2W2 + 13P3W1 + 8P3W2 + 9P3W3 + 14P4W2 + 8P4W3 + 5W1R1 + 6W1R2 + 7W1R3 + 10W1R4 + 12W2R3 + 8W2R4 + 10W2R5 + 14W2R6 + 14W3R4 + 12W3R5 + 12W3R6 + 6W3R7
ST
	P1W1 + P1W2 <= 150
	P2W1 + P2W2 <= 450
	P3W1 + P3W2 + P3W3 <= 250
	P4W2 + P4W3 <= 150
	W1R1 >= 100
	W1R2 >= 150
	W1R3 + W2R3 >= 100
	W1R4 + W2R4 + W3R4 >= 200
	W2R5 + W3R5 >= 200
	W2R6 + W3R6 >= 150
	W3R7 >= 100
	W1R1 + W1R2 + W1R3 + W1R4 - P1W1 - P2W1 - P3W1 = 0
	W2R3 + W2R4 + W2R5 + W2R6 - P1W2 - P2W2 - P3W2 - P4W2 = 0
	W3R4 + W3R5 + W3R6 + W3R7 - P3W3 - P4W3 = 0
	P1W1 >= 0
	P1W2 >= 0
	P2W1 >= 0
	P2W2 >= 0
	P3W1 >= 0
	P3W2 >= 0
	P3W3 >= 0
	P4W2 >= 0
	P4W3 >= 0
	W1R1 >= 0
	W1R2 >= 0
	W1R3 >= 0
	W1R4 >= 0
	W2R3 >= 0
	W2R4 >= 0
	W2R5 >= 0
	W2R6 >= 0
	W3R4 >= 0
	W3R5 >= 0
	W3R6 >= 0
	W3R7 >= 0
END
```

Copy of Code

```
min 10P1W1 + 15P1W2 + 11P2W1 + 8P2W2 + 13P3W1 + 8P3W2 + 9P3W3 + 14P4W2 + 8P4W3 + 5W1R1
+ 6W1R2 + 7W1R3 + 10W1R4 + 12W2R3 + 8W2R4 + 10W2R5 + 14W2R6 + 14W3R4 + 12W3R5 + 12W3R6
+ 6W3R7
ST
	P1W1 + P1W2 <= 150
	P2W1 + P2W2 <= 450
	P3W1 + P3W2 + P3W3 <= 250
	P4W2 + P4W3 <= 150
	W1R1 >= 100
	W1R2 >= 150
	W1R3 + W2R3  >= 100
	W1R4 + W2R4 + W3R4 >= 200
	W2R5 + W3R5 >= 200
	W2R6 + W3R6 >= 150
	W3R7 >= 100

	W1R1 + W1R2 + W1R3 + W1R4 - P1W1 - P2W1 - P3W1 = 0
	W2R3 + W2R4 + W2R5 + W2R6 - P1W2 - P2W2 - P3W2 - P4W2 = 0
	W3R4 + W3R5 + W3R6 + W3R7 - P3W3 - P4W3 = 0

	P1W1 >= 0
	P1W2 >= 0
	P2W1 >= 0
	P2W2 >= 0
	P3W1 >= 0
	P3W2 >= 0
	P3W3 >= 0
	P4W2 >= 0
	P4W3 >= 0
```

W1R1 >= 0
W1R2 >= 0
W1R3 >= 0
W1R4 >= 0
W2R3 >= 0
W2R4 >= 0
W2R5 >= 0
W2R6 >= 0
W3R4 >= 0
W3R5 >= 0
W3R6 >= 0
W3R7 >= 0
END

Result (Output)

The minimum cost is **$17,100**

Route	Quantity	Price per item	Cost
P1W1	150	10	1500
P2W1	200	11	2200
P2W2	250	8	2000
P3W2	150	8	1200
P3W3	100	9	900
P4W3	150	8	1200
W1R1	100	5	500
W1R2	150	6	900
W1R3	100	7	700
W2R4	200	8	1600
W2R5	200	10	2000
W3R6	150	12	1800
W3R7	100	6	600
Total	2000		17,100

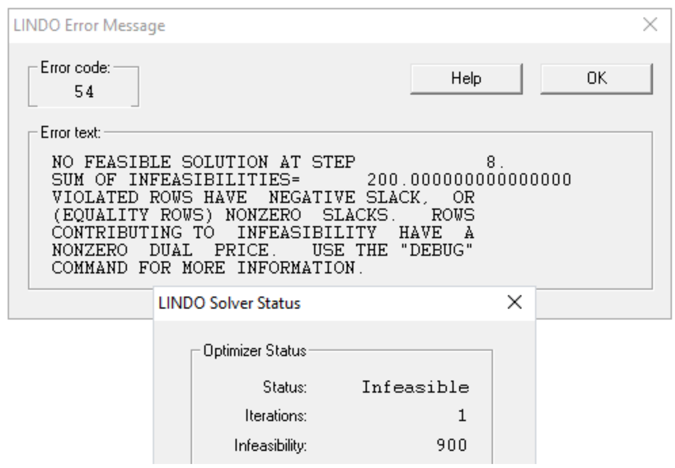
```
LP OPTIMUM FOUND AT STEP     13

      OBJECTIVE FUNCTION VALUE

      1)      17100.00

 VARIABLE        VALUE          REDUCED COST
     P1W1       150.000000          0.000000
     P1W2         0.000000          8.000000
     P2W1       200.000000          0.000000
     P2W2       250.000000          0.000000
     P3W1         0.000000          2.000000
     P3W2       150.000000          0.000000
     P3W3       100.000000          0.000000
     P4W2         0.000000          7.000000
     P4W3       150.000000          0.000000
     W1R1       100.000000          0.000000
     W1R2       150.000000          0.000000
     W1R3       100.000000          0.000000
     W1R4         0.000000          5.000000
     W2R3         0.000000          2.000000
     W2R4       200.000000          0.000000
     W2R5       200.000000          0.000000
     W2R6         0.000000          1.000000
     W3R4         0.000000          7.000000
     W3R5         0.000000          3.000000
     W3R6       150.000000          0.000000
     W3R7       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
```

3). B). Due to old infrastructure Warehouse 2 is going to close eliminating all of the associated routes. What is the optimal solution for this modified model? Is it feasible to ship all the refrigerators to either warehouse 1 or 3 and then to the retailers without using warehouse 2? Why or why not?

This modified model is not feasible or no solution to that because, according to the graph, there are lots of retailers like R5, R6, and R7 cannot be reached through warehouse 1 but only 3 if warehouse 2 is closed. Since P1 and P2 can only access to warehouse 1, those retailers cannot be reached without warehouse 2. It also breaks the constraint for R5 if the warehouse 2 is closed because if only P3 and P4 are able supply R5, R6, and R7 with their supply capacity of 400 in combined, they will not be enough to supply all those retailers. P3 and P4 have a combined supply capacity of 400 but R5, R6, and R7 have a combined demand of 450. Though P3 and P4 can supply R6 and R7 sufficiently, (400-250=150), that leaves the combined supply capacity with only 150 but R5 demands for 200, therefore without warehouse 2, P3 and P4 will not be able to supply the combined demand of R5, R6, and R7. We can also test this hypothesis by modify the code and run it on LINDO, the program resulted an error.

Here's the output result:

```
min 10P1W1 +  11P2W1  + 13P3W1  + 9P3W3 +  8P4W3 + 5W1R1 + 6W1R2 + 7W1R3 + 10W1R4 +  14W3R4 + 12W3R5 + 12W3R6 + 6W3R7
ST
        P1W1 <= 150
        P2W1  <= 450
        P3W1  + P3W3 <= 250
        P4W3 <= 150
        W1R1 >= 100
        W1R2 >= 150
        W1R3  >= 100
        W1R4 + W3R4 >= 200
        W3R5 >= 200
         W3R6 >= 150

        W1R1 + W1R2 + W1R3 + W1R4 - P1W1 - P2W1 - P3W1 = 0

        W3R4 + W3R5 + W3R6 + W3R7 - P4W3 = 0
        P1W1 >= 0

        P2W1 >= 0

        P3W1 >= 0

        P3W3 >= 0

        P4W3 >= 0
        W1R1 >= 0
        W1R2 >= 0
        W1R3 >= 0
        W1R4 >= 0

        W3R4 >= 0
        W3R5 >= 0
        W3R6 >= 0
        W3R7 >= 0
END
```



Copy of Code

min 10P1W1 + 11P2W1 + 13P3W1 + 9P3W3 + 8P4W3 + 5W1R1 + 6W1R2 + 7W1R3 + 10W1R4 + 14W3R4 + 12W3R5 + 12W3R6 + 6W3R7

ST

P1W1 <= 150

P2W1 <= 450

P3W1 + P3W3 <= 250

P4W3 <= 150

W1R1 >= 100

W1R2 >= 150

W1R3 >= 100

W1R4 + W3R4 >= 200

W3R5 >= 200

W3R6 >= 150

W3R7 >= 100

W1R1 + W1R2 + W1R3 + W1R4 - P1W1 - P2W1 - P3W1 = 0
W3R4 + W3R5 + W3R6 + W3R7 - P3W3 - P4W3 = 0

P1W1 >= 0

P2W1 >= 0

P3W1 >= 0

P3W3 >= 0

P4W3 >= 0
W1R1 >= 0
W1R2 >= 0
W1R3 >= 0
W1R4 >= 0

W3R4 >= 0
W3R5 >= 0
W3R6 >= 0
W3R7 >= 0

END

3). C). Instead of closing Warehouse 2 management has decide to keep a portion of it open but limit shipments to 100 refrigerators per week. Is this feasible? If so what is the optimal solution when warehouse 2 is limited to 100 refrigerators?

The modified model is feasible and the optimal solution for the minimized cost when warehouse 2 is limited to 100 refrigerators is 18,300. Here are the results for the modified model with LINDO:

LP Code

```
min 10P1W1 + 15P1W2 + 11P2W1 + 8P2W2 + 13P3W1 + 8P3W2 + 9P3W3 + 14P4W2 + 8P4W3 + 5W1R1 + 6W1R2 + 7W1R3 + 10W1R4 + 12W2R3 + 8W2R4 + 10W2R5 + 14W2R6 + 14W3R4 + 12W3R5 + 12W3R6 + 6W3R7
ST
        P1W1 + P1W2 <= 150
        P2W1 + P2W2 <= 450
        P3W1 + P3W2 + P3W3 <= 250
        P4W2 + P4W3 <= 150
        W1R1 >= 100
        W1R2 >= 150
        W1R3 + W2R3  >= 100
        W1R4 + W2R4 + W3R4 >= 200
        W2R5 + W3R5 >= 200
        W2R6 + W3R6 >= 150
        W3R7 >= 100

        W1R1 + W1R2 + W1R3 + W1R4 - P1W1 - P2W1 - P3W1 = 0
        W2R3 + W2R4 + W2R5 + W2R6 - P1W2 - P2W2 - P3W2 - P4W2 = 0
        W3R4 + W3R5 + W3R6 + W3R7 - P3W3 - P4W3 = 0

        P1W2 + P2W2 + P3W2 + P4W2 <= 100
        W2R3 + W2R4 + W2R5 + W2R6 <= 100

        P1W1 >= 0
        P1W2 >= 0
        P2W1 >= 0
        P2W2 >= 0
        P3W1 >= 0
        P3W2 >= 0
        P3W3 >= 0
        P4W2 >= 0
        P4W3 >= 0
        W1R1 >= 0
        W1R2 >= 0
        W1R3 >= 0
        W1R4 >= 0
        W2R3 >= 0
        W2R4 >= 0
        W2R5 >= 0
        W2R6 >= 0
        W3R4 >= 0
        W3R5 >= 0
        W3R6 >= 0
        W3R7 >= 0
END
```

Copy of Code

```
min 10P1W1 + 15P1W2 + 11P2W1 + 8P2W2 + 13P3W1 + 8P3W2 + 9P3W3 + 14P4W2 + 8P4W3 + 5W1R1
+ 6W1R2 + 7W1R3 + 10W1R4 + 12W2R3 + 8W2R4 + 10W2R5 + 14W2R6 + 14W3R4 + 12W3R5 + 12W3R6
+ 6W3R7
ST
        P1W1 + P1W2 <= 150
        P2W1 + P2W2 <= 450
        P3W1 + P3W2 + P3W3 <= 250
        P4W2 + P4W3 <= 150
        W1R1 >= 100
        W1R2 >= 150
        W1R3 + W2R3  >= 100
        W1R4 + W2R4 + W3R4 >= 200
        W2R5 + W3R5 >= 200
        W2R6 + W3R6 >= 150
        W3R7 >= 100

        W1R1 + W1R2 + W1R3 + W1R4 - P1W1 - P2W1 - P3W1 = 0
        W2R3 + W2R4 + W2R5 + W2R6 - P1W2 - P2W2 - P3W2 - P4W2 = 0
        W3R4 + W3R5 + W3R6 + W3R7 - P3W3 - P4W3 = 0

        P1W2 + P2W2 + P3W2 + P4W2 <= 100
        W2R3 + W2R4 + W2R5 + W2R6 <= 100

        P1W1 >= 0
```

P1W2 >= 0
P2W1 >= 0
P2W2 >= 0
P3W1 >= 0
P3W2 >= 0
P3W3 >= 0
P4W2 >= 0
P4W3 >= 0
W1R1 >= 0
W1R2 >= 0
W1R3 >= 0
W1R4 >= 0
W2R3 >= 0
W2R4 >= 0
W2R5 >= 0
W2R6 >= 0
W3R4 >= 0
W3R5 >= 0
W3R6 >= 0
W3R7 >= 0
END

Result (Output)

```
        OBJECTIVE FUNCTION VALUE

        1)      18300.00

  VARIABLE        VALUE          REDUCED COST
      P1W1       150.000000          0.000000
      P1W2         0.000000          8.000000
      P2W1       350.000000          0.000000
      P2W2       100.000000          0.000000
      P3W1         0.000000          4.000000
      P3W2         0.000000          2.000000
      P3W3       250.000000          0.000000
      P4W2         0.000000          9.000000
      P4W3       150.000000          0.000000
      W1R1       100.000000          0.000000
      W1R2       150.000000          0.000000
      W1R3       100.000000          0.000000
      W1R4       150.000000          0.000000
      W2R3         0.000000          7.000000
      W2R4        50.000000          0.000000
      W2R5        50.000000          0.000000
      W2R6         0.000000          4.000000
      W3R4         0.000000          4.000000
      W3R5       150.000000          0.000000
      W3R6       150.000000          0.000000
      W3R7       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)         0.000000         11.000000
       14)         0.000000          8.000000
       15)         0.000000         11.000000
       16)         0.000000          0.000000
       17)         0.000000          5.000000
       18)       150.000000          0.000000
       19)         0.000000          0.000000
       20)       350.000000          0.000000
       21)       100.000000          0.000000
       22)         0.000000          0.000000
       23)         0.000000          0.000000
       24)       250.000000          0.000000
       25)         0.000000          0.000000
       26)       150.000000          0.000000
       27)       100.000000          0.000000
       28)       150.000000          0.000000
       29)       100.000000          0.000000
       30)       150.000000          0.000000
       31)         0.000000          0.000000
       32)        50.000000          0.000000
       33)        50.000000          0.000000
       34)         0.000000          0.000000
       35)         0.000000          0.000000
       36)       150.000000          0.000000
       37)       150.000000          0.000000
       38)       100.000000          0.000000

 NO. ITERATIONS=      14
```

4). A). V = [1, 5, 10, 25] and A = 202.

Objective function is the min of V1 + V2 + V3 + V4. The constraints are they should be non-negatives and 1V1 + 5V2 + 10V3 + 25V4 = 202. The minimum number of coins for making a change is 10 with two coins of 1’s and eight coins of 25’s.

Copy of Code

```
min V1 + V2 + V3 + V4
ST
        1V1 + 5V2 + 10V3 + 25V4 = 202
        V1>=0
        V2>=0
        V3>=0
        V4>=0
END
GIN V1
GIN V2
GIN V3
GIN V4
```

Result (Output)

LP Code

```
min V1 + V2 + V3 + V4
ST
        1V1 + 5V2 + 10V3 + 25V4 = 202
        V1>=0
        V2>=0
        V3>=0
        V4>=0
END
GIN V1
GIN V2
GIN V3
GIN V4
```

```
      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
      3)         2.000000          0.000000
      4)         0.000000          0.000000
      5)         0.000000          0.000000
      6)         8.000000          0.000000

NO. ITERATIONS=        63
BRANCHES=    12 DETERM.=   1.000E    0
```

B). V = [1, 3, 7, 12, 27] and A = 293

Objective function is the min of V1 + V2 + V3 + V4 + V5. The constraints are they should be non-negatives and 1V1 + 3V2 + 7V3 + 12V4 + 27V5 = 293. The minimum number of coins for making a change is 14 with two coins of 7's, three coins of 12's, and nine coins of 27's.

Copy of Code

```
min V1 + V2 + V3 + V4 + V5
ST
        1V1 + 3V2 + 7V3 + 12V4 + 27V5 = 293
        V1>=0
        V2>=0
        V3>=0
        V4>=0
        V5>=0
END
GIN V1
GIN V2
GIN V3
GIN V4
GIN V5
```

Result (Output)

LP Code

```
min V1 + V2 + V3 + V4 + V5
ST
        1V1 + 3V2 + 7V3 + 12V4 + 27V5 = 293
        V1>=0
        V2>=0
        V3>=0
        V4>=0
        V5>=0
END
GIN V1
GIN V2
GIN V3
GIN V4
GIN V5
```

```
        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
        3)         0.000000          0.000000
        4)         0.000000          0.000000
        5)         2.000000          0.000000
        6)         3.000000          0.000000
        7)         9.000000          0.000000

 NO. ITERATIONS=      98
 BRANCHES=    34 DETERM.=  1.000E    0
```