

Homework 2 - Advanced LP & Network Flow Models

Adv. Analytics and Metaheuristics

Daniel Carpenter and Christopher Ferguson

February 2022

Contents

1 - Problem 1	2
1.1 Problems a and b	2
1.2 Model Assumptions and Overview	2
1.3 Mathematical Formulation	3
1.4 Code and Output	5
2 - Problem 2	6
3 - Problem 3	7
3.1 Model Overview	7
3.2 Mathematical Formulation	8
3.3 Code and Output	9
4 - Problem 4	11
4.1 Model Overview	11
4.2 Mathematical Formulation	13
4.3 Code and Output	14

1 - Problem 1

1.1 Problems a and b

- Below shows how we get to the answer of Problems a and b

Problem 1 (a): Calculating the Upper Bound for the Raisin Product

Total Raisins Calculated	1,240,000	
Less Raisin Dedicated Portion to Grade A	(930,000)	75% <- note this is the % towards A
Grade B Portion Implied*	310,000	25% <- "" B
Total Grade B Available	5,270,000	
% Grade B Used Towards Raisins, i.e. tier 8	5.88%	

Interpretation and Assumptions:

*There must be 5.88% (Total of 310,000) of grade B's total supply that are on tier 8, which raisins are able to use since it satisfies the minimum points.

For raisins, this implies that 25% of the raisin production will stem from Grade B (and 75% towards A)

For juice, this implies that 75% of the juice production will stem from Grade B (and 25% towards A)

Note 100% Grade B used for Jelly.

Problem 1(b): Calculate the Fruit Cost seen in Table 3

Grade	Cost per Pound	
Grade A	\$ 0.45	<- noted in table 3 bullet 4
Grade B	\$ 0.25	

	Raisins	Juice	Jelly	
Pounds per Product	6.5	14.0	18.0	<- Noted in Table 2 footer

% Breakdown of Grades				
Grade A	75.0%	25.0%	0.0%	<- this is from the aboves assumptions text
Grade B	25.0%	75.0%	100.0%	<- 1 minus above
Weighted Avg. Cost per Pound	\$ 0.40	\$ 0.30	\$ 0.25	
Fruit Cost (Pounds * Wt. Avg. Cost)	\$ 2.60	\$ 4.20	\$ 4.50	<- this is how they determine the fruit cost in Table 3

1.2 Model Assumptions and Overview

- This model calculates the maximum net profit for a set of products and inputs grades (See below sets).

1.3 Mathematical Formulation

1.3.1 Sets

Set Name	Description
<i>FRUIT</i>	Fruit grade levels of grapes, which are Grade A and Grade B grapes
<i>PRODUCTS</i>	Types of products to be sold, which are Raisins , Juice , and Jelly

1.3.2 Parameters

Parameter Name	Description
$amountOfFruit_f$	<i>Pounds</i> of fruit grade ($f \in FRUIT$) available to use
$avgGradeOfFruit_f$	Avg. point quality of fruit grade ($f \in FRUIT$)
$productLimit_p$	Upper bound of number of products ($p \in PRODUCTS$) to produce
$poundsPerProduct_p$	The amount of pounds associated with a single unit of product ($p \in PRODUCTS$)
$contrToProfit_p$	The contribution to profit of product ($p \in PRODUCTS$)
$netProfit_p$	The net profit (<i>net of OH allocation</i>) of product ($p \in PRODUCTS$)
$productGradeLimit_p$	Requirement of mean point quality of a product ($p \in PRODUCTS$)

1.3.3 Decision Variables

Variable Name	Description
$produce_{f,p}$	Number (<i>as integer</i>) of products ($p \in PRODUCTS$) to be produced by fruit grade ($f \in FRUIT$)

1.3.4 Objective Function

$$\text{Maximize Net_Profit : } \sum_{f \in FRUIT, p \in PRODUCTS} produce_{f,p} \times netProfit_p$$

1.3.5 Constraints

C1: For each fruit grade ($f \in FRUIT$), the number of products produced ($p \in PRODUCTS$) must be equal to numbers of fruit provided

$$maxWeight : \sum_{p \in PRODUCTS} (produce_{f,p} \times numbersPerProduct_p) = amountOfFruit_f, \forall f \in FRUIT$$

C2: For each product produced ($p \in PRODUCTS$), limit the number of products p produced to \leq to the demanded (product numbers)

$$demand : \sum_{f \in FRUIT} produce_{f,p} \leq productLimit_p, \forall p \in PRODUCTS$$

C3: For each product ($p \in PRODUCTS$), the grade of fruit ($f \in FRUIT$) is greater than the minimum required grade of the product ($p \in PRODUCTS$).

minAvgGrade :

$$\sum_{f \in FRUIT} (produce_{f,p} \times avgGradeOfFruit_f) \geq productGradeLimit_p \times \sum_{f \in FRUIT} (produce_{f,p}),$$

$$\forall p \in PRODUCTS$$

C4: Non-negativity constraints

$$produce_{f,p} \geq 0, \forall f \in FRUIT, \forall p \in PRODUCTS$$

1.4 Code and Output

1.4.1 Code

```
# group12_HW2_plmod.m X
C:\Users\danielcarpenter> OneDrive - The Chickasaw Nation\Documents\GITHUB\OU-DSA\Metaheuristics\03-Homework\HW02\AMPL\Models\group12_1
7
8 # Reset globals
9 options solver cplex; # Using cplex for simplex alg
10
11 # SETS =====
12 set FRUIT circular; # Circular to index the kth element for constraint 2
13 set PRODUCTS circular; # ""
14
15 # PARAMETERS =====
16 param amountOffruit {FRUIT} >= 0; # Pounds of fruit available by grade
17 param avgGradeOffruit {FRUIT}; # Avg fruit grade
18 param productLimit {PRODUCTS} >= 0; # Demand for products
19 param poundsPerProduct {PRODUCTS}; # Pounds associated with a product
20 param contrToProfit {PRODUCTS}; # Product contribution to profit
21 param netProfit {PRODUCTS}; # Net profit per product
22 param productGradeLimit {PRODUCTS}; # The min grade for each product
23
24 # DECISION VARIABLES =====
25 # Include upper and lower bounds
26 var produce {f in FRUIT, p in PRODUCTS} >= 0 integer;
27 # var grade {p in PRODUCTS} >= 0, <= 10;
28
29 # OBJECTIVE FUNCTION =====
30 maximize Total_Profit:
31 sum {f in FRUIT, p in PRODUCTS} produce[f,p] * netProfit[p];
32
33 # CONSTRAINTS =====
34
35 ## C1: Number of pounds of the products produced (p in PRODUCTS) must be <= pounds of fruit
36 ## provided (f in FRUIT)
37 subject to maxWeight {f in FRUIT}:
38 (sum {p in PRODUCTS} produce[f,p] * poundsPerProduct[p]) == amountOffruit[f];
39
40 ## C2: Limit the number of products to <= the demand
41 subject to demand {p in PRODUCTS}:
42 (sum {f in FRUIT} produce[f,p]) <= productLimit[p];
43
44 ## C3: For all products (p in PRODUCTS), the weighted-average grade of fruit is greater than the
45 ## minimum required grade
46 subject to minAvgGrade {p in PRODUCTS}:
47 (sum {f in FRUIT} produce[f,p] * avgGradeOffruit[f])
48 >= productGradeLimit[p] * (sum {f in FRUIT} produce[f,p]);
49
50 # CONTROLS =====
51 data group12_HW2_pl.dat;
52 solve;
53
54 print;
55 print "At maximum profit, the number of products to produce (by fruit grade)";
56 display produce;
57
```

1.4.2 Output

CPLEX 20.1.0.0: optimal integer solution within mipgap or absmipgap; objective 107593.36
14 MIP simplex iterations
0 branch-and-bound nodes
absmipgap = 2.60826, relmipgap = 2.42418e-05

At maximum profit, the number of products to produce (by fruit grade):

```
produce :=
gradeA raisins      74996
gradeA juice        31609
gradeA jelly         0
gradeB raisins      24988
gradeB juice        94827
gradeB jelly        210000
;
```

Optimal profit is \$107,600. 74,996 Raisins produced using Grade A, 24,988 of Grade B, and so on for each product. See validation of the weighted-average point constraint below.

		Grade A	Grade B							
Avg Grade of Fruit		9	5							
Product	Grade A	Grade B	Pounds per Product	Equation using the produce var			Product Grade Limit	Calculated Weighted Avg Points by Product		
				LHS	RHS	inequality Holds?		Weight A	Weight B	
Raisin	74,996	24,988	7	799,904	>=	799,872	TRUE	8	75%	25%
Juice	31,609	94,827	14	758,616	>=	758,616	TRUE	6	25%	75%
Jelly	-	210,000	18	1,050,000	>=	-	TRUE	0	0%	100%
Total per Grade	930,000	5,270,000								
Total Lbs		6,200,000	Equation: $SUM_f (produce_{fp} * avgGradeOffruit_f) >= productGradeLimit_p * SUM_f (produce_{fp})$, for all p in PRODUCTS							
		Calculated Net Profit								
		Net Profit	Profit							
		0.35	34,994.40							
		(0.14)	(17,701.04)							
		0.43	90,300.00							
		\$107,593.36								

2 - Problem 2

3 - Problem 3

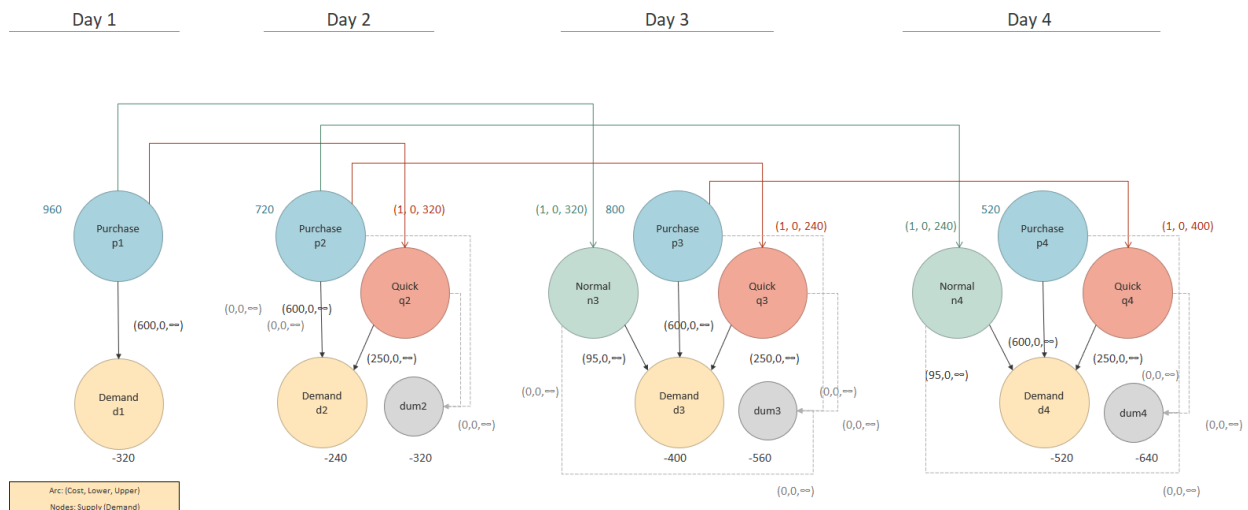
3.1 Model Overview

3.1.1 Assumptions and Calculations for Network Flow Diagram

- Below shows how we decided to balance the network with supply and dummy nodes
- In order to obtain a balanced network (i.e. supply equals demand), we must allow for all possible routes to have enough supply.
- Excess supply allowed on certain days is captured by dummy nodes so that the business does not actually produce the tires.

Determine Supply and Dummy Allocation					
Day	Demand (d)	Incoming Arcs (in)	Potential Supply $in * d$	Dummy Allocation $d_{day-1} + d_{day-2}$	
1	-320	3	960	0	
2	-240	3	720	-320	
3	-400	2	800	-560	
4	-520	1	520	-640	

3.1.2 Network Flow Diagram



3.2 Mathematical Formulation

3.2.1 Sets, Parameters, Decision Vars

Set Name	Description
<i>NODES</i>	Set of all nodes in above network flow diagram:
p1 p2 p3 p4	Number of tires to <i>purchase</i> on day $\in(1-4)$
n1 n2 n3 n4	Number of tires to reshape using the <i>normal</i> service on day $\in(1-4)$
q1 q2 q3 q4	Number of tires to reshape using the <i>quick</i> service on day $\in(1-4)$
d1 d2 d3 d4	Demand of tires on each day $\in(1-4)$
dum2 dum3 dum4	Dummy nodes to balance excess supply for days 2 3 and 4 $\in(2-4)$. Do not need one for day 1 since purchasing

The set A is a set of *arcs*, e.g. (i, j) for $i \in N, j \in N$ each of which may carry *flow of a commodity*

Decision variable: x_{ij} determines the units of flow on arc (i, j)

Arc (i, j)

- cost c_{ij} per unit of flow on arc (i, j)
- upper bound on flow of u_{ij} (capacity)
- lower bound on flow of ℓ_{ij} (usually 0)

3.2.2 Objective, and Constraints

$$\begin{aligned}
 &\text{minimize} && \sum_{(i,j) \in A} c_{ij} x_{ij} \\
 &\text{subject to} && \sum_{j:(i,j) \in A} x_{ij} - \sum_{j:(j,i) \in A} x_{ji} = b_i \quad \forall i \in N \\
 &&& \ell_{ij} \leq x_{ij} \leq u_{ij} \quad \forall (i, j) \in A
 \end{aligned}$$

- Upper and lower bounds use to direct the flow of tires from *purchasing* to *quick* or *normal* service

3.3 Code and Output

3.3.1 Model: group12_HW2_p3.mod

- Used mcnpf.txt from course website and renamed to group12_HW2_p3.mod.
- Added data group12_HW2_p3.dat; solve; and display x;

3.3.2 Data: group12_HW2_p3.dat

```
5 # Problem 4
6
7 data;
8
9 # Set of Nodes for each day (1,2,3,4)
10 set NODES := p1 p2 p3 p4 # Purchases
11             n1 n2 n3 n4 # Normal Service
12             q1 q2 q3 q4 # Quick Service
13             d1 d2 d3 d4 # Demand
14             dum2 dum3 dum4 # Dummy for days 2 3 and 4
15 ;
16
17 # Set of ARCS from Node i to Node j
18 set ARCS :=
19             (p1, *) d1 q2 n3
20             (p2, *) d2 q3 n4 dum2
21             (p3, *) d3 q4 dum3
22             (p4, dum4)
23             (q2, *) d2 dum2
24             (q3, *) d3 dum3
25             (q4, *) d4 dum4
26             (n3, *) d3 dum3
27             (n4, *) d4 dum4
28 ;
29
30 # Number of Products demanded
31 param b :=
32     # "Supply" nodes. Not actually supply but needs to have origin
33     p1 960
34     p2 720
35     p3 800
36     p4 520
37
38     # Transshipment
39     n1 0 n2 0 n3 0 n4 0
40     q1 0 q2 0 q3 0 q4 0
41
42     # Dummy Nodes to take excess supply
43     dum2 -320 # Dif between d2 and p2
44     dum3 -560 # ""3
45     dum4 -640
46
47     # Demand nodes
48     d1 -320
49     d2 -240
50     d3 -400
51     d4 -520
52 ;
53
```

Data Continued:

```

54 # The cost (c), lower (l), and upper (u), from node i to j in IN NODES
55 param:      c      l      u :=
56     [p1, d1] 600    .    .
57     [p2, d2] 600    .    .
58     [p3, d3] 600    .    .
59     [q2, d2] 250    .    .
60     [q3, d3] 250    .    .
61     [q4, d4] 250    .    .
62     [n3, d3] 95     .    .
63     [n4, d4] 95     .    .
64     [p1, q2] 1      .    320
65     [p1, n3] 1      .    320
66     [p2, q3] 1      .    240
67     [p2, n4] 1      .    240
68     [p3, q4] 1      .    400
69     [p2, dum2] 0     .    .
70     [p3, dum3] 0     .    .
71     [p4, dum4] 0     .    .
72     [q2, dum2] 0     .    .
73     [q3, dum3] 0     .    .
74     [q4, dum4] 0     .    .
75     [n3, dum3] 0     .    .
76     [n4, dum4] 0     .    .
77 ;
78

```

3.3.3 Output

- Total minimized cost: 396,720
- Interpretation of the tires purchased on each day:
 1. 320 tires purchased
 2. 240 tires reshaped with Quick Service from previous day
 3. 80 Reshaped with quick service from previous day. 320 tires used from reshaping via Normal service from day 1.
 4. 280 Reshaped with quick service from previous day. 240 tires used from reshaping via Normal service from day 2.

```

ampl: model 'C:\Users\daniel.carpenter\OneDrive - the Chickasaw Na
CPLEX 20.1.0.0: optimal solution; objective 396720
1 dual simplex iterations (0 in phase I)
x [*,*] (tr)
:      n3      n4      p1      p2      p3      p4      q2      q3      q4      :=
d1      .      .      320      .      .      .      .      .      .
d2      .      .      .      0      .      .      240      .      .
d3      320      .      .      .      0      .      .      80      .
d4      .      240      .      .      .      .      .      .      280
dum2      .      .      .      240      .      .      80      .      .
dum3      0      .      .      .      400      .      160      .      .
dum4      .      0      .      .      .      520      .      .      120
n3      .      .      320      .      .      .      .      .      .
n4      .      .      .      240      .      .      .      .      .
q2      .      .      320      .      .      .      .      .      .
q3      .      .      .      240      .      .      .      .      .
q4      .      .      .      .      400      .      .      .      .
;

```

4 - Problem 4

4.1 Model Overview

4.1.1 Assumptions and Calculations for Network Flow Diagram

- Goal of below tables are to put all data on a *per unit of product* basis
- Need to be on per unit basis so that we can effectively minimize the cost
- *Color of tables* correspond to the network nodes on the next page

Labor, Manufacturing, and Transportation Cost Calculations for Arcs

Labor (Cost per Unit Output and Total Supply Available)					
Type	Cost per Person	Unit Output per Person	Cost / Unit	Total Labor Avail	TTL Product Supply
Specialist	\$ 2,000	12	\$ 166.67	100	1,200
Generalist	\$ 1,700	10	\$ 170.00	200	2,000

Cost of Transportation			
	Scranton, PA	Utica, NY	Stamford, CT
Per Person	300	250	275
Per Unit of Product (trans. Cost / unit output by type)			
Specialist	\$ 25.00	\$ 20.83	\$ 22.92
Generalist	\$ 30.00	\$ 25.00	\$ 27.50

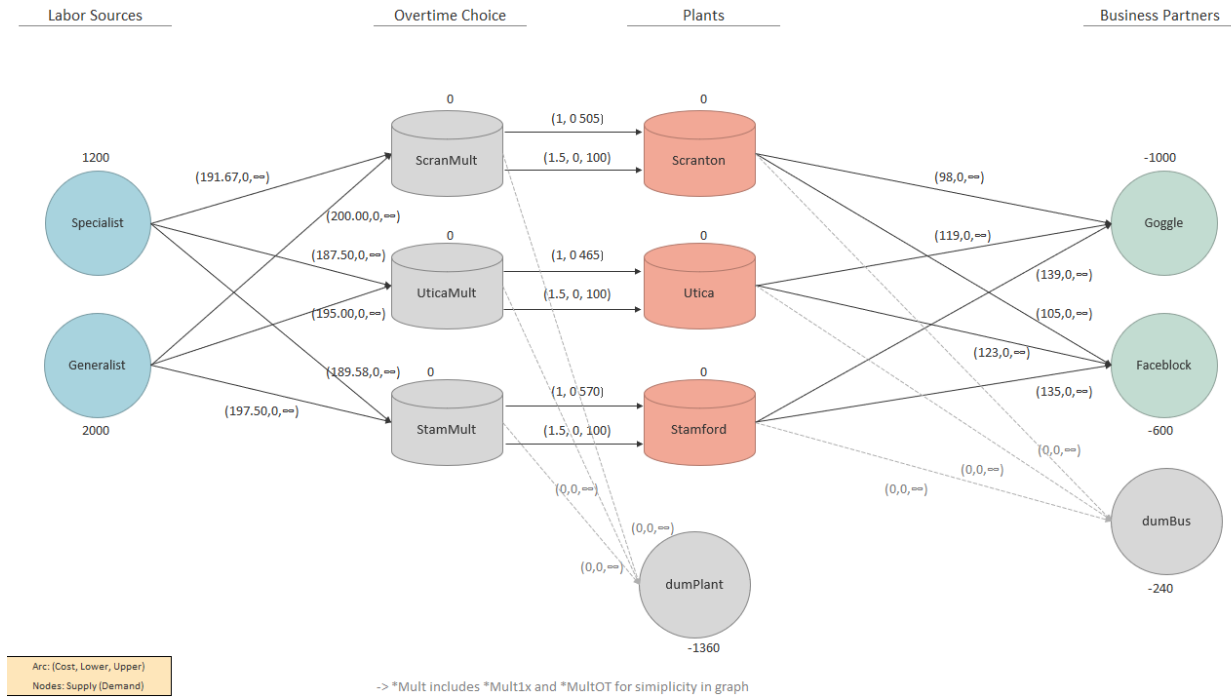
Cost of Transportation + Labor per Unit of Output			
	Scranton, PA	Utica, NY	Stamford, CT
Specialist	\$ 191.67	\$ 187.50	\$ 189.58
Generalist	\$ 200.00	\$ 195.00	\$ 197.50

Plant Production Limits			
	Base	OT	OT Mult
Scranton	505	100	1.5
Utica	465	100	1.5
Stamford	570	100	1.5

Manufacturing and Transportation Costs			
	Manufacture	Goggle	Faceblock
Scranton	\$ 90	\$ 8	\$ 15
Utica	\$ 105	\$ 14	\$ 18
Stamford	\$ 115	\$ 24	\$ 20

Man. + Trans Cost	
Goggle	Faceblock
\$ 98	\$ 105
\$ 119	\$ 123
\$ 139	\$ 135

4.1.2 Network Flow Diagram



4.2 Mathematical Formulation

4.2.1 Sets, Parameters, Decision Vars

Set Name	Description
<i>NODES</i>	Set of all nodes in above network flow diagram:
Specialist, Generalist	The two types of Supply of Labor
ScranMult1x, UticaMult1x, StamMult1x	Passed through if <i>did not</i> use overtime
ScranMultOT, UticaMultOT, StamMultOT	Passed through if <i>did</i> use overtime
Scranton, Utica, Stamford	Transshipment nodes which are the plants
dumPlant, dumBus	Dummy nodes that account for excess supply from unbalanced supply from labor nodes

The set A is a set of *arcs*, e.g. (i, j) for $i \in N, j \in N$
each of which may carry *flow of a commodity*

Decision variable: x_{ij} determines the units of flow on arc (i, j)

Arc (i, j)

- cost c_{ij} per unit of flow on arc (i, j)
- upper bound on flow of u_{ij} (capacity)
- lower bound on flow of ℓ_{ij} (usually 0)

4.2.2 Objective, and Constraints

$$\begin{aligned}
 &\text{minimize} && \sum_{(i,j) \in A} c_{ij} x_{ij} \\
 &\text{subject to} && \sum_{j:(i,j) \in A} x_{ij} - \sum_{j:(j,i) \in A} x_{ji} = b_i \quad \forall i \in N \\
 &&& \ell_{ij} \leq x_{ij} \leq u_{ij} \quad \forall (i, j) \in A
 \end{aligned}$$

- Upper and lower bounds use to direct the flow of the product

4.3 Code and Output

4.3.1 Model: group12_HW2_p4.mod

- Used mcnpf.txt from course website and renamed to group12_HW2_p4.mod.
- Added data group12_HW2_p4.dat; solve; and display x;

4.3.2 Data: group12_HW2_p4.dat

```
group12_HW2_p4.mod M  group12_HW2_p4.dat M
C:\Users\daniel.carpenter> OneDrive - the Chickasaw Nation > Documents > GitHub > OU-DSA > Metaheuristics > 03 - Homework > HW 02 > AMPL Models > gr
1  # Homework 2 - Advanced LP & Network Flow Models
2  # Adv. Analytics and Metaheuristics
3  # Daniel Carpenter and Christopher Ferguson
4  # February 2022
5  # Problem 4
6
7  data;
8
9  # Set of Nodes containing all sources of Labor, Plants, and Business Partners
10 set NODES :=    Specialist Generalist          # Labor
11                ScrantMult1x UticaMult1x StamMult1x # Multiplier dumBus 1x no OT
12                ScrantMultOT UticaMultOT StamMultOT # Multiplier dumBus 1.5x for OT
13                dumPlant                          # Dummy to limit supply before plants
14                Scranton Utica Stamford           # Plants
15                Goggle Facebook dumBus            # Business Partners + dumBus for unbalanced supply/demand
16                ;
17
18 # Set of ARCS from Labor Sources, to Plants, to Business Partners
19 set ARCS :=
20     # No OT used
21     (Specialist, *) ScrantMult1x UticaMult1x StamMult1x
22     (Generalist, *) ScrantMult1x UticaMult1x StamMult1x
23     (ScrantMult1x, *) Scranton dumPlant
24     (UticaMult1x, *) Utica dumPlant
25     (StamMult1x, *) Stamford dumPlant
26
27     # If Use OT
28     (Specialist, *) ScrantMultOT UticaMultOT StamMultOT
29     (Generalist, *) ScrantMultOT UticaMultOT StamMultOT
30     (ScrantMultOT, *) Scranton dumPlant
31     (UticaMultOT, *) Utica dumPlant
32     (StamMultOT, *) Stamford dumPlant
33
34     # Plants to demanders and dumBus for unbalanced network
35     (Scranton, *) Goggle Facebook dumBus
36     (Utica, *) Goggle Facebook dumBus
37     (Stamford, *) Goggle Facebook dumBus
38     ;
39
40 # Number of Products demanded by Business Partner
41 param b :=
42     # Supply
43     Specialist 1200 # 12 units * 100 Labor hours
44     Generalist 2000 # 10 units * 200 Labor hours
45
46     # Transshipment (Plants or OT Multiplier)
47     ScrantMult1x 0 # Not using OT
48     UticaMult1x 0
49     StamMult1x 0
50     ScrantMultOT 0 # Using OT
51     UticaMultOT 0
52     StamMultOT 0
53     Scranton 0 # Plant Arrival
54     Utica 0
55     Stamford 0
56
```

Data Continued:

```

group12_HW2_p4.mod M  group12_HW2_p4.dat M
C: > Users > daniel.carpenter > OneDrive - the Chickasaw Nation > Documents > GitHub > OU-DSA > Metaheuristics > 03 - Homework > HW 02 > AMPL M
57      # Demand
58      dumPlant    -1360  # (505+465+570 + 3*100) - (1200+2000)
59      Goggle      -1000
60      Faceblock   -600
61      dumBus      -240  # (-1000-600) - (-1360)
62      ;
63
64      # The cost (c), lower (l), and upper (u), from node i to j in IN NODES
65      param:                c      l      u :=
66      # Cost of Transportation + Labor per Unit of Output (see screenshot)
67      # PATH **NOT** USING OT -----
68      [Specialist, ScrantMult1x] 191.67 . . # Supply -> Mult
69      [Specialist, UticaMult1x] 187.5 . .
70      [Specialist, StamMult1x] 189.58 . .
71      [Generalist, ScrantMult1x] 200 . .
72      [Generalist, UticaMult1x] 195 . .
73      [Generalist, StamMult1x] 197.5 . .
74      [ScranMult1x, Scranton] 1 . 505 # Mult -> Plant
75      [UticaMult1x, Utica] 1 . 465
76      [StamMult1x, Stamford] 1 . 570
77      [ScranMult1x, dumPlant] 0 . . # Mult -> Dummy
78      [UticaMult1x, dumPlant] 0 . .
79      [StamMult1x, dumPlant] 0 . .
80
81      # PATH USING OT -----
82      [Specialist, ScrantMultOT] 191.67 . . # Supply -> Mult
83      [Specialist, UticaMultOT] 187.5 . .
84      [Specialist, StamMultOT] 189.58 . .
85      [Generalist, ScrantMultOT] 200 . .
86      [Generalist, UticaMultOT] 195 . .
87      [Generalist, StamMultOT] 197.5 . .
88      [UticaMultOT, Utica] 1.5 . 100 # Mult -> Plant
89      [ScranMultOT, Scranton] 1.5 . 100
90      [StamMultOT, Stamford] 1.5 . 100
91      [ScranMultOT, dumPlant] 0 . . # Mult -> Dummy
92      [UticaMultOT, dumPlant] 0 . .
93      [StamMultOT, dumPlant] 0 . .
94
95      # Plant to Business Partners
96      [Scranton, Goggle] 98 . . # Plant -> Demanders
97      [Scranton, Faceblock] 105 . .
98      [Scranton, dumBus] 0 . .
99      [Utica, Goggle] 119 . .
100     [Utica, Faceblock] 123 . .
101     [Utica, dumBus] 0 . .
102     [Stamford, Goggle] 139 . .
103     [Stamford, Faceblock] 135 . .
104     [Stamford, dumBus] 0 . .
105     ;
106

```

4.3.3 Output

- Total minimized cost: \$806,192.95
- Scranton, Utica, and Stamford produce 0, 430, and 170 units of product for **Facebook**, respectively.
- Scranton, Utica, and Stamford produce 605, 0, and 395 units of product for **Goggle**, respectively.
- All possible products produced (using a portion of the available regular and overtime hours). 200 products produced by Specialists using OT, and 100 from generalists using OT.

```

ampl: model 'C:\Users\daniel.carpenter\OneDrive - the Chickasaw Nation\Documents\Githu
CPLEX 20.1.0.0: optimal solution; objective 806192.95
13 dual simplex iterations (0 in phase I)
x [*,*] (tr)
# $1 = Generalist
# $2 = ScranMult1x
# $3 = ScranMultOT
# $4 = Scranton
# $5 = Specialist
# $6 = StamMult1x
# $7 = StamMultOT
# $8 = Stamford
# $10 = UticaMult1x
# $11 = UticaMultOT
:
      $1      $2      $3      $4      $5      $6      $7      $8      Utica      $10      $11 :=
Facebook      .      .      .      0      .      .      .      430      170      .      .
Goggle      .      .      .      605      .      .      .      0      395      .      .
ScranMult1x      0      .      .      .      505      .      .      .      .      .      .
ScranMultOT      0      .      .      .      100      .      .      .      .      .      .
Scranton      .      505      100      .      .      .      .      .      .      .      .
StamMult1x      75      .      .      .      495      .      .      .      .      .      .
StamMultOT      0      .      .      .      100      .      .      .      .      .      .
Stamford      .      .      .      .      .      570      100      .      .      .      .
Utica      .      .      .      .      .      .      .      .      .      465      100
UticaMult1x      1825      .      .      .      0      .      .      .      .      .      .
UticaMultOT      100      .      .      .      0      .      .      .      .      .      .
dumBus      .      .      .      0      .      .      .      240      0      .      .
dumPlant      .      0      0      .      .      0      0      .      .      1360      0
;

```