Homework 2 - Advanced LP & Network Flow Models

Adv. Analytics and Metaheuristics

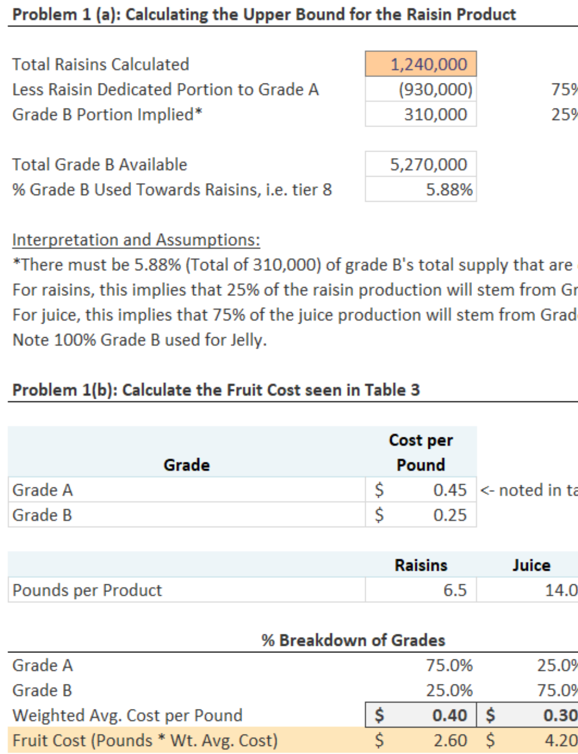
Daniel Carpenter and Christopher Ferguson

February 2022

Contents

1 - Problem 1

1.1 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%
Grade B Portion Implied*	310,000	25%
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
For raisins, this implies that 25% of the raisin production will stem from Gr
For juice, this implies that 75% of the juice production will stem from Grad
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 ta
Grade B	$ 0.25	

	Raisins	Juice
Pounds per Product	6.5	14.0
% Breakdown of Grades		
Grade A	75.0%	25.0%
Grade B	25.0%	75.0%
Weighted Avg. Cost per Pound	$ 0.40	$ 0.30
Fruit Cost (Pounds * Wt. Avg. Cost)	$ 2.60	$ 4.20

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

1.2 Mathematical Formulation

1.2.1 Sets

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

1.2.2 Parameters

Parameter Name	Description
$amountOfFruit_f$	*Tons* of fruit grade ($f \in FRUIT$) available to use

Parameter Name	Description
$avgGradeOfFruit_f$	Avg. point quality of fruit grade ($f \in FRUIT$)
$productLimit_p$	Upper bound of production of product ($p \in PRODUCTS$) *in tons*
$poundsPerProduct_p$	The amount of pounds associated with a single unit of product ($p \in PRODUCTS$)
$contrToProfit_p$	The contribution to profit *per ton* of product ($p \in PRODUCTS$)
$netProfit_p$	The net profit (*net of OH allocation*) *per ton* of product ($p \in PRODUCTS$)
$productGradeLimit_p$	Requirement of mean point quality of a product ($p \in PRODUCTS$)

1.2.3 Decision Variables

Variable Name	Description
$produce\ _{f,p}$	The *tons* of products ($p \in PRODUCTS$) to be produced by fruit grade ($f \in FRUIT$)
$grade_p$	The point quality of each product ($p \in PRODUCTS$) created

1.2.4 Objective Function

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

1.2.5 Constraints

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

$$maxTons : \sum_{p \in PRODUCTS} produce_{f,p} = amountOfFruit_f, \ \forall \ f \in FRUIT$$

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

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

C3: Cannot produce `Jelly` using `Grade A`

$$noGradeAJelly : produce_{f,p} = 0, \ \ for \ f \in \text{Grade A}, \ \ for \ p \in \text{Jelly}$$

CX: Non-negativity constraints, and max point quality is 10

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

$$0 \leq grade_p \leq \ 10, \ \forall \ f \in FRUIT, \ \forall \ p \in PRODUCTS$$

2 - Problem 2

3 - Problem 3

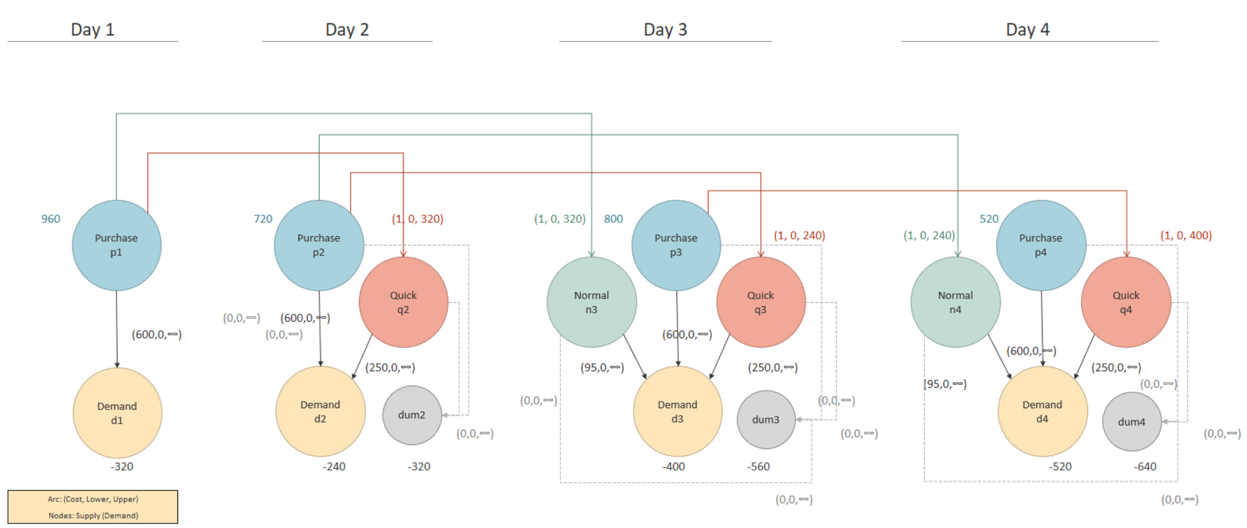
3.1 Model Overview

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
NODES	Set of all nodes in above network flow diagram:
`p1 p2 p3 p4`	Number of tires to *purchase* on day ∈(1-4)
`n1 n2 n3 n4`	Number of tires to reshape using the *normal* service on day ∈(1-4)
`q1 q2 q3 q4`	Number of tires to reshape using the *quick* service on day ∈(1-4)
`d1 d2 d3 d4`	Demand of tires on each day ∈(1-4)
`dum2 dum3 dum4`	Dummy nodes to balance excess supply for days 2 3 and 4 ∈(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

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

- 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 `mcnfp.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`

```
# Problem 4

data;

# Set of Nodes for each day (1,2,3,4)
set NODES :=    p1 p2 p3 p4     # Purchases
                n1 n2 n3 n4     # Normal Service
                q1 q2 q3 q4     # Quick Service
                d1 d2 d3 d4     # Demand
                dum2 dum3 dum4  # Dummy for days 2 3 and 4
                ;

# Set of ARCS from Node i to Node j
set ARCS :=
            (p1, *) d1 q2 n3
            (p2, *) d2 q3 n4 dum2
            (p3, *) d3 q4    dum3
            (p4, dum4)
            (q2, *) d2       dum2
            (q3, *) d3       dum3
            (q4, *) d4       dum4
            (n3, *) d3       dum3
            (n4, *) d4       dum4
            ;

# Number of Products demanded
param b :=
    # "Supply" nodes. Not actually supply but needs to have origin
    p1 960
    p2 720
    p3 800
    p4 520

    # Transshippment
    n1 0 n2 0 n3 0 n4 0
    q1 0 q2 0 q3 0 q4 0

    # Dummy Nodes to take excess supply
    dum2 -320 # Dif between d2 and p2
    dum3 -560 # ""3
    dum4 -640

    # Demand nodes
    d1   -320
    d2   -240
    d3   -400
    d4   -520
    ;
```

Data Continued:

```
# The cost (c), lower (l). and upper (u), from node i to j in IN NODES
param:      c       l   u :=
    [p1, d1] 600    .   .
    [p2, d2] 600    .   .
    [p3, d3] 600    .   .
    [q2, d2] 250    .   .
    [q3, d3] 250    .   .
    [q4, d4] 250    .   .
    [n3, d3] 95     .   .
    [n4, d4] 95     .   .
    [p1, q2] 1      .   320
    [p1, n3] 1      .   320
    [p2, q3] 1      .   240
    [p2, n4] 1      .   240
    [p3, q4] 1      .   400
    [p2, dum2] 0    .   .
    [p3, dum3] 0    .   .
    [p4, dum4] 0    .   .
    [q2, dum2] 0    .   .
    [q3, dum3] 0    .   .
    [q4, dum4] 0    .   .
    [n3, dum3] 0    .   .
    [n4, dum4] 0    .   .
    ;
```

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

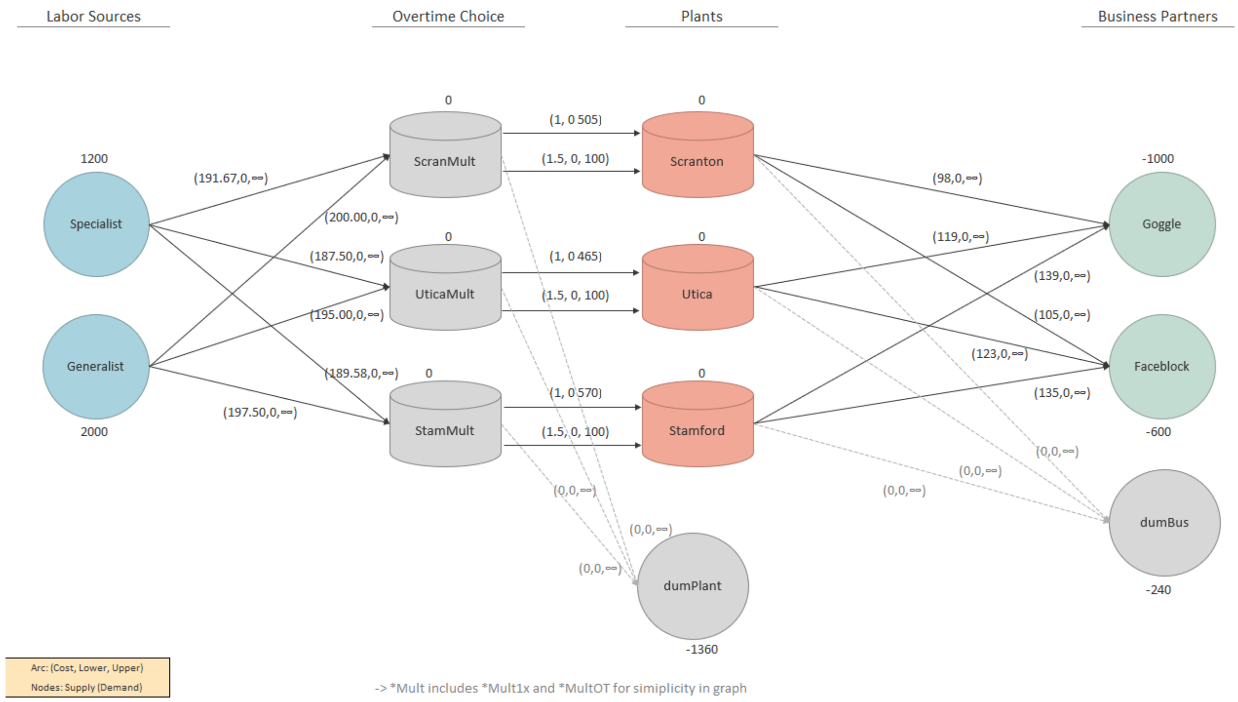
Cost of Transportation + Labor per Unit of Output						
	Scranton, PA	Utica, NY	Stamford, CT	Scranton, PA	Utica, NY	Stamford, CT
Specialist	$ 191.67	$ 187.50	$ 189.58	(191.67,0,∞)	(187.50,0,∞)	(189.58,0,∞)
Generalist	$ 200.00	$ 195.00	$ 197.50	(200.00,0,∞)	(195.00,0,∞)	(197.50,0,∞)

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
NODES	Set of all nodes in above network flow diagram:
`Specialist`, `Generalist`	The two types of Supply of Labor
`ScranMult1x`, `UticaMult1x`, `StamMult1x`	Passed through if *did not* use overtime
`ScranMultOT`, `UticaMultOT`, `StamMultOT`	Passed through if *did* 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

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

- 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 `mcnfp.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
```

```
# Homework 2 - Advanced LP & Network Flow Models
# Adv. Analytics and Metaheuristics
# Daniel Carpenter and Christopher Ferguson
# February 2022
# Problem 4

data;

# Set of Nodes containing all sources of Labor, Plants, and Business Partners
set NODES :=    Specialist Generalist                # Labor
                ScranMult1x UticaMult1x StamMult1x # Multiplier dumBus 1x no OT
                ScranMultOT UticaMultOT StamMultOT # Multiplier dumBus 1.5x for OT
                dumPlant                           # Dummy to limit supply before plants
                Scranton Utica Stamford            # Plants
                Goggle Faceblock dumBus            # Business Partners + dumBus for unbalanced supply/demand
                ;

# Set of ARCS from Labor Sources, to Plants, to Business Partners
set ARCS :=
            # No OT used
            (Specialist,  *) ScranMult1x UticaMult1x StamMult1x
            (Generalist,  *) ScranMult1x UticaMult1x StamMult1x
            (ScranMult1x, *) Scranton dumPlant
            (UticaMult1x, *) Utica    dumPlant
            (StamMult1x,  *) Stamford dumPlant

            # If Use OT
            (Specialist, *) ScranMultOT UticaMultOT StamMultOT
            (Generalist, *) ScranMultOT UticaMultOT StamMultOT
            (ScranMultOT,*) Scranton dumPlant
            (UticaMultOT,*) Utica    dumPlant
            (StamMultOT, *) Stamford dumPlant

            # Plants to demanders and dumBus for unbalanced network
            (Scranton,  *) Goggle Faceblock dumBus
            (Utica,     *) Goggle Faceblock dumBus
            (Stamford,  *) Goggle Faceblock dumBus
            ;

# Number of Products demanded by Business Partner
param b :=
    # Supply
    Specialist  1200    # 12 units * 100 labor hours
    Generalist  2000    # 10 units * 200 labor hours

    # Transshippment (Plants or OT Multiplier)
    ScranMult1x 0  # Not using OT
    UticaMult1x 0
    StamMult1x  0
    ScranMultOT 0   # Using OT
    UticaMultOT 0
    StamMultOT  0
    Scranton    0   # Plant Arrival
    Utica       0
    Stamford    0
```

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

```
    # Demand
    dumPlant     -1360    # (505+465+570 + 3*100) - (1200+2000)
    Goggle       -1000
    Faceblock    -600
    dumBus       -240     # (-1000-600) - (-1360)
    ;

# The cost (c), lower (l). and upper (u), from node i to j in IN NODES
param:                        c       l   u :=
    # Cost of Transportation + Labor per Unit of Output (see screenshot)
    # PATH **NOT** USING OT ------------------------------
    [Specialist,  ScranMult1x] 191.67  .   .  # Supply -> Mult
    [Specialist,  UticaMult1x] 187.5   .   .
    [Specialist,  StamMult1x]  189.58  .   .
    [Generalist,  ScranMult1x] 200     .   .
    [Generalist,  UticaMult1x] 195     .   .
    [Generalist,  StamMult1x]  197.5   .   .
    [ScranMult1x, Scranton]    1       .   505 # Mult -> Plant
    [UticaMult1x, Utica]       1       .   465
    [StamMult1x,  Stamford]    1       .   570
    [ScranMult1x, dumPlant]    0       .   .   # Mult -> Dummy
    [UticaMult1x, dumPlant]    0       .   .
    [StamMult1x,  dumPlant]    0       .   .

    # PATH USING OT --------------------------------------
    [Specialist,  ScranMultOT] 191.67  .   .   # Supply -> Mult
    [Specialist,  UticaMultOT] 187.5   .   .
    [Specialist,  StamMultOT]  189.58  .   .
    [Generalist,  ScranMultOT] 200     .   .
    [Generalist,  UticaMultOT] 195     .   .
    [Generalist,  StamMultOT]  197.5   .   .
    [UticaMultOT, Utica]       1.5     .   100 # Mult -> Plant
    [ScranMultOT, Scranton]    1.5     .   100
    [StamMultOT,  Stamford]    1.5     .   100
    [ScranMultOT, dumPlant]    0       .   .   # Mult -> Dummy
    [UticaMultOT, dumPlant]    0       .   .
    [StamMultOT,  dumPlant]    0       .   .

    # Plant to Business Partners
    [Scranton,   Goggle]       98      .   .   # Plant -> Demanders
    [Scranton,   Faceblock]    105     .   .
    [Scranton,   dumBus]       0       .   .
    [Utica,      Goggle]       119     .   .
    [Utica,      Faceblock]    123     .   .
    [Utica,      dumBus]       0       .   .
    [Stamford,   Goggle]       139     .   .
    [Stamford,   Faceblock]    135     .   .
    [Stamford,   dumBus]       0       .   .
    ;
```

4.3.3 Output

- Total minimized cost: `$806,192.95`
- Scranton, Utica, and Stamford produce `0`, `430`, and `170` units of product for **Faceblock**, 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 :=
Faceblock        .     .     .     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
;
```