Homework 2 - Advanced LP & Network Flow Models

Adv. Analytics and Metaheuristics

Daniel Carpenter and Christopher Ferguson

February 2022

Contents

1 - Problem 1

2 - Problem 2

3 - Problem 3

3.1 Mathematical Formulation

3.1.1 Parameters

Parameter Name	Description	Unit
NUM_DAYS	Total number of days available, (e.g. 4)	
P	Price of *purchasing* new tires	Dollars per Tire
N	Cost of *normal* service on used tires	Dollars per Tire
Q	Cost of *quick* service on used tires	Dollars per Tire
r_j	Demand of tires	Tires per day ($j \in Days$)

3.1.2 Decision Variables

Variable Name	Description
$purch_j$	Number of tires to `purchase` on day ($j \in Days$)
$norm_j$	Number of tires to *reshape* using the `normal` service on day ($j \in Days$)
$quick_j$	Number of tires to *reshape* using the `quick` service on day ($j \in Days$)

3.1.3 Objective Function

- Minimize total cost by when using the three types of tire services

$$Minimize\ cost: \sum_{j}^{NUM\_DAYS} (purch_j \times P) \ + \ (norm_j \times N) \ + \ (quick_j \times Q)$$

3.1.4 Constraints

C1: Purchase everything on the first day

$$purchaseDay1 : purch_1 = r_1$$

C2: Can't use normal service on first 2 days *(Since day 1 = purchase, then takes a day to reshape w/Normal)*

$$normStartsAtDay2 : norm_j = 0,\ for\ j\ in\ 1, 2$$

C3: Can only reshape all tires from a full 24-hour day ago *(not yesterday's tires since > 24 hrs.)*

$$normLag : norm_j \leq purch_{j-2} + norm_{j-2} + quick_{j-2},\ for\ j\ in\ 3, \ldots, NUM\_DAYS$$

C4: Meet Daily Demand r_j

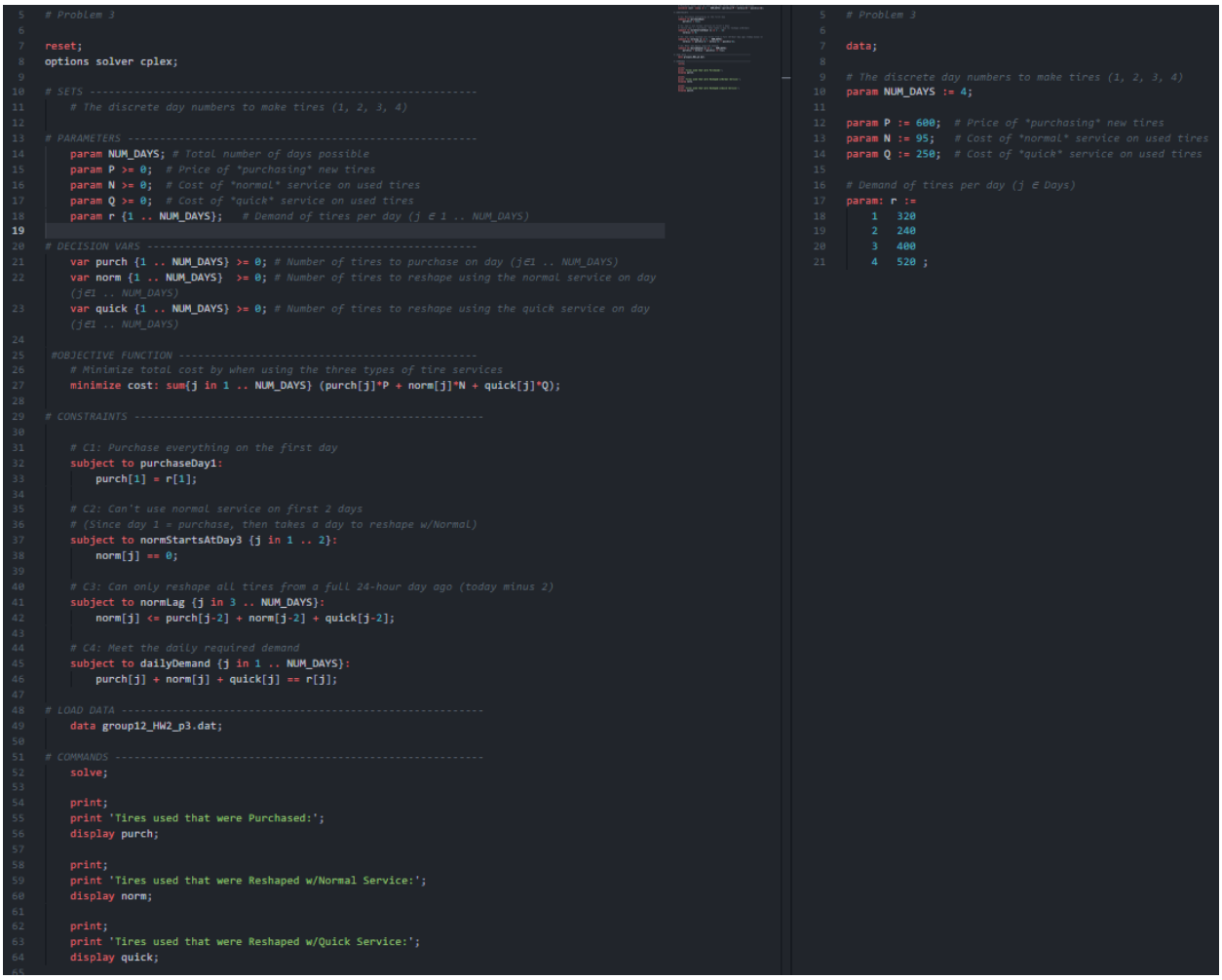
$$dailyDemand : \sum_{j}^{NUM\_DAYS} [purch_j\ +\ norm_j\ +\ quick_j] = r_j$$

C5: Non-Negativity Constraints

$$r_j, purch_j, norm_j, quick_j \geq 0$$

3.2 Code and Output

3.2.1 Code



```
# Problem 3

reset;
options solver cplex;

# SETS ------------------------------------------------------------
    # The discrete day numbers to make tires (1, 2, 3, 4)

# PARAMETERS ------------------------------------------------------
    param NUM_DAYS; # Total number of days possible
    param P >= 0;  # Price of *purchasing* new tires
    param N >= 0;  # Cost of *normal* service on used tires
    param Q >= 0;  # Cost of *quick* service on used tires
    param r {1 .. NUM_DAYS};   # Demand of tires per day (j ∈ 1 .. NUM_DAYS)

# DECISION VARS ---------------------------------------------------
    var purch {1 .. NUM_DAYS} >= 0; # Number of tires to purchase on day (j∈1 .. NUM_DAYS)
    var norm {1 .. NUM_DAYS}  >= 0; # Number of tires to reshape using the normal service on day
    (j∈1 .. NUM_DAYS)
    var quick {1 .. NUM_DAYS} >= 0; # Number of tires to reshape using the quick service on day
    (j∈1 .. NUM_DAYS)

 #OBJECTIVE FUNCTION ---------------------------------------------
    # Minimize total cost by when using the three types of tire services
    minimize cost: sum{j in 1 .. NUM_DAYS} (purch[j]*P + norm[j]*N + quick[j]*Q);

# CONSTRAINTS -----------------------------------------------------

    # C1: Purchase everything on the first day
    subject to purchaseDay1:
        purch[1] = r[1];

    # C2: Can't use normal service on first 2 days
    # (Since day 1 = purchase, then takes a day to reshape w/Normal)
    subject to normStartsAtDay3 {j in 1 .. 2}:
        norm[j] == 0;

    # C3: Can only reshape all tires from a full 24-hour day ago (today minus 2)
    subject to normLag {j in 3 .. NUM_DAYS}:
        norm[j] <= purch[j-2] + norm[j-2] + quick[j-2];

    # C4: Meet the daily required demand
    subject to dailyDemand {j in 1 .. NUM_DAYS}:
        purch[j] + norm[j] + quick[j] == r[j];

# LOAD DATA -------------------------------------------------------
    data group12_HW2_p3.dat;

# COMMANDS --------------------------------------------------------
    solve;

    print;
    print 'Tires used that were Purchased:';
    display purch;

    print;
    print 'Tires used that were Reshaped w/Normal Service:';
    display norm;

    print;
    print 'Tires used that were Reshaped w/Quick Service:';
    display quick;
```

```
# Problem 3

data;

# The discrete day numbers to make tires (1, 2, 3, 4)
param NUM_DAYS := 4;

param P := 600;  # Price of *purchasing* new tires
param N := 95;   # Cost of *normal* service on used tires
param Q := 250;  # Cost of *quick* service on used tires

# Demand of tires per day (j ∈ Days)
param: r :=
    1   320
    2   240
    3   400
    4   520 ;
```

3.2.2 Output

- Total minimized cost total 395,200
- Interpretation of each day:

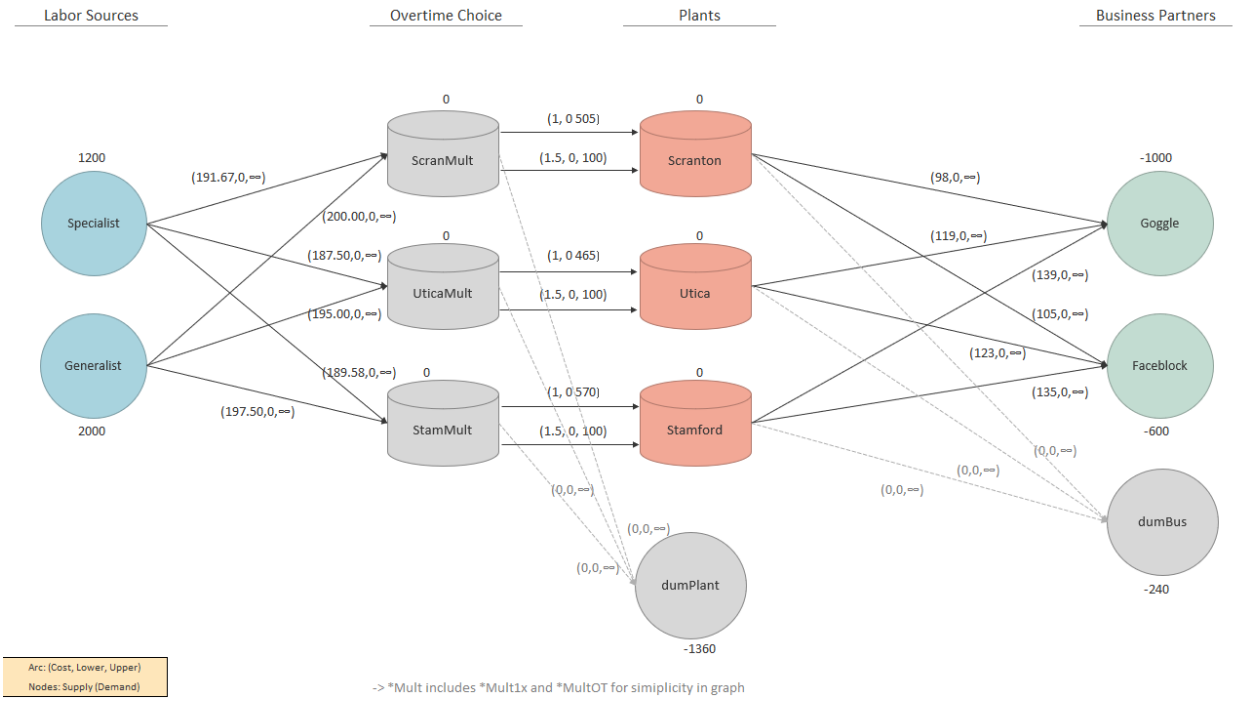
    1. 320 tires purchased
    2. 240 tires reshaped with Quick Service from previous day
    3. 80 Reshaped with quick service from previous day. 320 tires used from reshaping via Normal service from day 1.
    4. 280 Reshaped with quick service from previous day. 240 tires used from reshaping via Normal service from day 2.

```
ampl: model 'C:\Users\daniel.carpenter\OneDrive - the Chi
CPLEX 20.1.0.0: optimal solution; objective 395200
0 dual simplex iterations (0 in phase I)

Tires used that were Purchased:
purch [*] :=
1  320
2    0
3    0
4    0
;


Tires used that were Reshaped w/Normal Service:
norm [*] :=
1    0
2    0
3  320
4  240
;


Tires used that were Reshaped w/Quick Service:
quick [*] :=
1    0
2  240
3   80
4  280
;
```

4 - Problem 4

4.1 Model Overview

4.1.1 Network Flow Diagram



4.1.2 Calculations for Network Flow Diagram

Goal of below tables are to put all data on a *per unit of product* basis

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

4.2 Mathematical Formulation

4.2.1 Sets, Parameters, Decision Vars

Set Name	Description
$NODES$	Set of all nodes in above network flow diagram: Specialist, Generalist, ScranMult1x, UticaMult1x, StamMult1x, ScranMultOT, UticaMultOT, StamMultOT, dumPlant, Scranton, Utica, Stamford

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

$$\text{minimize} \sum_{(i,j)\in A} c_{ij}x_{ij}$$

$$\text{subject to} \sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = b_i \quad \forall i \in N$$

$$l_{ij} \leq x_{ij} \leq u_{ij} \qquad \forall (i,j) \in A$$

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

Shows the Objective, as well the flow from Node i to Node j for all Arcs:

- Total minimized cost: $806,192.95
- Each Column shows the source node and how much it sent to the destination node (which is shown for each row.)
- The values show the flow from source to destination (reference flow diagram for comparison)

```
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
;
```