ISE 4623/5023: Deterministic Systems Models / Systems Optimization

University of Oklahoma
School of Industrial and Systems Engineering
Fall 2021

Problem 1 (40 points): Transportation problem

Tiara company has two plants in different locations around the country where they produce drilling bits. Their sales partner has three distribution centers where they ship these drilling bits to their various customers. The plants can produce a given number of bits per week and the expected demand for each distribution center is also known. There is a shipping cost from each plant to each warehouse. Which plant should produce and ship how many bits to which distribution centers to meet the demand at each location with minimum cost? The following table shows the transportation cost from each plant to distribution centers as well as each distribution centers' demand and plants' supply.

Table 1. Transportation costs, per pound, from production plants to warehouses

From/To	Center 1	Center 2	Center 3	Plants' supply
Plant 1	$25	$85	$25	350
Plant 2	$50	$35	$95	300
Centers' demand	100	350	200	

a (15 points): Formulate the objective function and the necessary constraint using the given-information.

Parameters:
c_{ij}: Transportation cost for each product going from plant i to a center j
s_i: maximum supply capacity of the plant i
d_j: minimum demand of center j that needs to be satisfied

Variable:
x_{ij}: the number of products going from plant i to center j

Objective function:

$$Min\ z = \sum_i \sum_j x_{ij}$$

Constraints:

$$\sum_j x_{ij} = s_i \qquad \forall i \in I$$

$$\sum_i x_{ij} = d_j \qquad \forall j \in J$$

$$x_{ij} \geq 0$$

b (15 points): What is the transportation strategy (number of each product shipped from each plant to each warehouse, and from each warehouse to each retailer) that minimizes the total cost? To do that solve the problem using Excel or Gurobi. Show your results clearly and discuss.

```
from gurobipy import *
Plants = [1, 2]

Centers = [1, 2, 3]

cost = {
    (1, 1): 25,
    (1, 2): 85,
    (1, 3): 25,
    (2, 1): 50,
    (2, 2): 35,
    (2, 3): 95,
}

supply = {
    1: 350,
    2: 300
}

demand = {
    1: 100,
    2: 350,
    3: 200
}

# Create a new model
m = Model("transport_problem_1")

# Create variables

Flow = m.addVars(cost.keys(), name="Flow")

obj = sum(Flow[i] * cost[i] for i in cost)

# Add supply constraints
for p in Plants:
    m.addConstr(sum(Flow[(p, c)] for c in Centers) == supply[p])

# Add demand constraints
for c in Centers:
    m.addConstr(sum(Flow[(p, c)] for p in Plants) == demand[c])

# Optimize the model. The default ModelSense is to is to minimize the objective.
m.setObjective(obj, GRB.MINIMIZE)
m.optimize()
m.printAttr('X')

Solved in 0 iterations and 0.00 seconds
Optimal objective  2.225000000e+04

    Variable            X
-------------------------
   Flow[1,1]          100
   Flow[1,2]           50
   Flow[1,3]          200
   Flow[2,2]          300
```

c (10 points): Reformulate the problem for the case that each plant loses 30% of their production due to disruption. Solve the problem using Excel or Gurobi.

c_{ij}: Transportation cost for each product going from plant i to a center j
s_i: maximum supply capacity of the plant i
d_j: minimum demand of center j that needs to be satisfied

Variable:
x_{ij}: the number of products going from plant i to center j
δ_j: the centers unmet demad

Objective function:

$$Min\ \ z = \sum_i \sum_j x_{ij} + \sum_j M\delta_j$$

Constraints:

$$\sum_j x_{ij} = (1 - 0.3)s_i \qquad \forall i\ \in I$$

$$\sum_i x_{ij} = d_j - \delta_j \qquad \forall j\ \in J$$

$$x_{ij}\ \geq 0, \delta_j \geq 0$$

```
Plants = [1, 2]

Centers = [1, 2, 3]

cost = {
    (1, 1): 25,
    (1, 2): 85,
    (1, 3): 25,
    (2, 1): 50,
    (2, 2): 35,
    (2, 3): 95,

}

supply = {
    1: 350,
    2: 300
}

demand = {
    1: 100,
    2: 350,
    3: 200
}


# Create a new model
m = Model("transport_problem_1")

# Create variables

Flow = m.addVars(cost.keys(), name="Flow")
unmet = m.addVars(Centers, name="unmet")
s = m.addVar(name='s')

obj = sum(Flow[i] * cost[i] for i in cost) + 1000*sum(unmet[j] for j in Centers)

m. addConstr(s == sum(Flow[i] * cost[i] for i in cost))

# Add supply constraints
for p in Plants:
    m.addConstr(sum(Flow[(p, c)] for c in Centers) == supply[p]*(1-0.3))

# Add demand constraints
for c in Centers:
    m.addConstr(sum(Flow[(p, c)] for p in Plants) == demand[c] - unmet[c])

# Optimize the model. The default ModelSense is to is to minimize the objective.
m.setObjective(obj, GRB.MINIMIZE)
m.optimize()
m.printAttr('X')

Solved in 3 iterations and 0.00 seconds
Optimal objective  2.084750000e+05

    Variable            X
-------------------------
   Flow[1,1]           45
   Flow[1,3]          200
   Flow[2,2]          210
    unmet[1]           55
    unmet[2]          140
           s        13475
```

Problem 2 (60 points mandatory for everyone + 30 points for graduate students/extra credit): Minimum cost flow problem

Part 1 (60 points): Single commodity supply chain transportation

Transportation plays a significant role in a supply chain and its efficiency. Reducing the transportation costs between the manufacturers, warehouses, and retailers helps to provide more affordable product prices for customers. A major manufacturing company, MC Inc., produces its star products, Product A, to be sold at five retailers across the state. First, the products are manufactured in one of three plants, then distributed to three warehouses for storage, and finally are distributed to each of the retailers. Each retailer sends a separate order to the manufacture's head office, which is then dispatched from the appropriate warehouse to the retailer. The company has collected data regarding weekly orders and production and would like to find a way to minimize the costs of the entire operation.

The objective in this multicommodity supply chain problem is to minimize the total cost of transportation and unmet demand. Following tables provide information related to the transportation cost from production to warehouses, and from warehouses to retailers (Table 2-3).

Table 2. Transportation costs, per pound, from production plants to warehouses

	Unit shipping cost (per pound)		
From/To	Warehouse 1	Warehouse 2	Warehouse 3
Plant 1	$25	$85	$25
Plant 2	$50	$35	$95
Plant 3	$50	$40	$55

Table 3. Transportation costs, per pound, from warehouses to retailers

	Unit shipping cost (per pound)				
From/To	Retailer 1	Retailer 2	Retailer 3	Retailer 4	Retailer 5
Warehouse A	$75	$50	$60	$75	$30
Warehouse B	$85	$15	$85	$85	$90
Warehouse C	$90	$85	$35	$35	$95

One of the most important obstacles for the company is the work force performance. The number of workers in Plant 1, Plant 2, and Plan 3 are 90, 120, 80, respectively. Also, assume that each employee works 48 hours per week. Furthermore, Table 4 illustrates the amount of units that can be produced by a worker every hour.

Table 4. Average number of units produced per person per hour

Production facility	Product A (units/hour-person)
Plant 1	0.5
Plant 2	0.5
Plant 3	0.2

Tip: generally, in a network there are three different types of nodes: (i) supply nodes, (ii) transshipment nodes, and (iii) demand nodes. Note that, commodities arrive and depart the transshipment nodes, however, in demand nodes commodities are going to be consumed by the nodes. For these nodes, limited capacities may cause unmet demands, which should be accounted for (of course, given that you want to satisfy the demands as much as possible, you would need to penalize any unsupplied demand).

Table 5. Weekly orders (in number of units) made by the retailers on the first week of October

Product request	Retailer 1	Retailer 2	Retailer 3	Retailer 4	Retailer 5
Product A	175	120	140	100	160

There is also limitation for sending material from plants to warehouses and from warehouses to retailers. Meaning that only certain weight of products can be sent through every link. Table 6 and 7 show the necessary information regarding the required formulation.

Table 6. Maximum weekly weight (of products) that can be sent from each plant to each warehouse

	Maximum weekly amount of product shipped (in pounds)		
From/To	Warehouse 1	Warehouse 2	Warehouse 3
Plant 1	120	150	170
Plant 2	150	160	180
Plant 3	150	170	180

Table 7. Maximum weekly weight (of products) that can be sent from each warehouse to each retailer

	Maximum weekly amount of product shipped (in pounds)				
From/To	Retailer 1	Retailer 2	Retailer 3	Retailer 4	Retailer 5
Warehouse A	160	190	110	180	150
Warehouse B	170	190	150	140	120
Warehouse C	140	160	180	120	100

a (30 points): Formulate the necessary objective function and constraints required to minimize the transportation and unmet demand cost, using the aforementioned information defining the required variables and parameters.

Parameters:

ϕ_i: average number of units produced per worker per hour from plant $i \in P$ of product $l \in L$

q_i: number of workers in plant $i \in P$

d_i: demand in each retailer ($i \in R$). Note that this could also be used to model the demand in each warehouse ($i \in W$) , which is zero, as it is a transshipment node

c_{ij}: transportation cost, per pound, associated with arc $(i, j) \in A$

u_{ij}: maximum flow capacity, in pounds, associated with arc $(i, j) \in A$

e : weight, in pounds, of product A
o: number of weekly hours worked per worker in each plant
M: cost/penalty per unit of unmet demand

Variables:
x_{ij}: the number of units of product through arc $(i, j) \in A$
z_i: unmet demand of product in retailer $i \in R$

Objective function:

$$Min \sum_{(i,j)\in A} ec_{ij}x_{ij} + \sum_{i\in R} M\,z_i$$

Constraints:

$$\sum_{j:(i,j)\in A} x_{ij}/\phi_i \leq oq_i \qquad \forall i \in P$$

$$\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ij} = -d_i + z_i \qquad \forall i \in R$$

$$\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ij} = 0 \qquad \forall i \in W$$

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

$$x_{ij} \geq 0, z_i > 0$$

b (30 points): What is the transportation strategy (number of each product shipped from each plant to each warehouse, and from each warehouse to each retailer) that minimizes the total cost? Also, you need to report total unmet demand as well. To do that solve the problem using Excel or Gurobi. Show your results clearly and discuss.

```
from gurobipy import *

# Sets
N = ["p1", "p2", "p3", "w1", "w2", "w3", "r1", "r2", "r3", "r4", "r5"]
P = ["p1", "p2", "p3"]
W = ["w1", "w2", "w3"]
R = ["r1", "r2", "r3", "r4", "r5"]
L = ["A", "B"]

#PorameterS

A, u = multidict({
  ("p1", "w1"): 120,
  ("p1", "w2"): 150,
  ("p1", "w3"): 170,
  ("p2", "w1"): 150,
  ("p2", "w2"): 160,
  ("p2", "w3"): 180,
  ('p3', "w1"): 150,
  ("p3", "w2"): 170,
  ("p3", "w3"): 180,
  ("w1", "r1"): 160,
```

```python
  ("w1", "r2"): 190,
  ("w1", "r3"): 110,
  ("w1", "r4"): 180,
  ("w1", "r5"): 150,
  ("w2", "r1"): 170,
  ("w2", "r2"): 190,
  ("w2", "r3"): 150,
  ("w2", "r4"): 140,
  ("w2", "r5"): 120,
  ("w3", "r1"): 140,
  ('w3', "r2"): 160,
  ("w3", "r3"): 180,
  ("w3", "r4"): 120,
  ("w3", "r5"): 100})

c = {
  ("p1", "w1"): 25,
  ("p1", "w2"): 85,
  ("p1", "w3"): 25,
  ("p2", "w1"): 50,
  ("p2", "w2"): 35,
  ("p2", "w3"): 95,
  ('p3', "w1"): 50,
  ("p3", "w2"): 40,
  ("p3", "w3"): 55,
  ("w1", "r1"): 75,
  ("w1", "r2"): 50,
  ("w1", "r3"): 60,
  ("w1", "r4"): 75,
  ("w1", "r5"): 30,
  ("w2", "r1"): 85,
  ("w2", "r2"): 15,
  ("w2", "r3"): 85,
  ("w2", "r4"): 85,
  ("w2", "r5"): 90,
  ("w3", "r1"): 90,
  ('w3', "r2"): 85,
  ("w3", "r3"): 35,
  ("w3", "r4"): 35,
  ("w3", "r5"): 95}


e = 1


d = {

  ("w1"): 0,
  ("w2"): 0,
  ("w3"): 0,
  ("r1"): 175,
  ("r2"): 120,
  ("r3"): 140,
  ("r4"): 100,
  ("r5"): 160,

}


phi = {
    ("p1"): 0.5,
    ("p2"): 0.5,
    ("p3"): 0.2,
}

q = {
    ("p1"): 90,
    ("p2"): 120,
    ("p3"): 80,
}

o = 48
m = 1000

model = Model()
model.__len__ = 1
x = model.addVars(A, vtype=GRB.INTEGER, name='x')
z = model.addVars(R, vtype=GRB.INTEGER, name="z")

obj1 = sum(c[i, j]*e*x[i, j] for i, j in A) + sum(m*z[i] for i in R)

model.addConstrs((((sum(x[i, j]/phi[i] for j in N if (i, j) in A)) <= o*q[i]) for i in P), "c1")

model.addConstrs((x.sum(i, '*')-x.sum('*', i) == -d[i]+z[i] for i in R), "c2")
```

```
model.addConstrs((x.sum(i, '*')-x.sum('*', i) == -d[i] for i in W), "c3")

model.addConstrs((e*x[i, j] <= u[i, j] for (i, j) in A), "c5")

model.setObjective(obj1, GRB.MINIMIZE)

model.optimize()
model.printAttr('X')
model.write('jigigoo.lp')

Optimal solution found (tolerance 1.00e-04)
Best objective 5.447500000000e+04, best bound 5.447500000000e+04, gap 0.0000%
    Variable            X
-------------------------
    x[p1,w1]          120
    x[p1,w3]          170
    x[p2,w2]          160
    x[p3,w1]           35
    x[p3,w2]          140
    x[p3,w3]           70
    x[w1,r1]            5
    x[w1,r5]          150
    x[w2,r1]          170
    x[w2,r2]          120
    x[w2,r5]           10
    x[w3,r3]          140
```

Part 2 (20 points): (Bonus for undergraduates/mandatory for graduates) Multi commodity supply chain transportation

To expand their market share, the company decided to fabricate a second product, Product B. To fabricate both Product A and Product B, all plants were adapted accordingly. The workers' performance related to the Product B has been illustrated in Table 8.

Table 8. Average number of units produced per person per hour

Production facility	Product B (units/hour-person)
Plant 1	0.1
Plant 2	0.1
Plant 3	0.4

Note that the weekly orders for Product B required by the retailers are mentioned in Table 9.

Table 9. Weekly orders (in number of units) made by the retailers on the first week of October

Product request	Retailer 1	Retailer 2	Retailer 3	Retailer 4	Retailer 5
Product B	100	150	110	300	230

Keep in mind that one unit of Product 1 weighs one pound, while one unit of Product B weighs 3 pounds.

a (10 points): Consider having multi commodity in supply chain (Product A, and B), reformulate the problem to minimize the transportation and unmet demand cost.

Parameters:
ϕ_{il}: average number of units produced per worker per hour from plant $i \in P$ of product $l \in L$
q_i: number of workers in plant $i \in P$

d_{il}: demand in each retailer ($i \in R$). Note that this could also be used to model the demand in each warehouse ($i \in W$) , which is zero, as it is a transshipment node
c_{ij}: transportation cost, per pound, associated with arc $(i, j) \in A$
u_{ij}: maximum flow capacity, in pounds, associated with arc $(i, j) \in A$
e_l: weight, in pounds, of product $l \in L$
o: number of weekly hours worked per worker in each plant
M: cost/penalty per unit of unmet demand

Variables:
x_{ijl}: the number of units of product through arc $(i, j) \in A$
z_{il}: unmet demand of product in retailer $i \in R$

Objective function:

$$Min \sum_{(i,j)\in A} \sum_{l\in L} e_l c_{ij} x_{ijl} + \sum_{i\in R} \sum_{l\in L} M\, z_{il}$$

Constraints:

$$\sum_{j:(i,j)\in A} \sum_{l\in L} x_{ijl}/\phi_{il} \le o q_i \qquad \forall i \in P$$

$$\sum_{j:(i,j)\in A} x_{ijl} - \sum_{j:(j,i)\in A} x_{ijl} = -d_{il} + z_{il} \qquad \forall i \in R, \forall l \in L$$

$$\sum_{j:(i,j)\in A} x_{ijl} - \sum_{j:(j,i)\in A} x_{ijl} = 0 \qquad \forall i \in W, \forall l \in L$$

$$\sum_{l\in L} e_l x_{ijl} \le u_{ij} \qquad \forall (i,j) \in A$$

$$x_{ijl} \ge 0, \qquad z_{il} \ge 0$$

b (10 points): Solve the problem using Excel or Gurobi to find the optimal transportation strategy (number of each product shipped from each plant to each warehouse, and from each warehouse to each retailer) with respect to total cost minimization. Show your results clearly and discuss.

```
from gurobipy import *

# Sets
N = ["p1", "p2", "p3", "w1", "w2", "w3", "r1", "r2", "r3", "r4", "r5"]
P = ["p1", "p2", "p3"]
W = ["w1", "w2", "w3"]
R = ["r1", "r2", "r3", "r4", "r5"]
L = ["A", "B"]

#Parameters
```

```python
A, u = multidict({
  ("p1", "w1"): 120,
  ("p1", "w2"): 150,
  ("p1", "w3"): 170,
  ("p2", "w1"): 150,
  ("p2", "w2"): 160,
  ("p2", "w3"): 180,
  ('p3', "w1"): 150,
  ("p3", "w2"): 170,
  ("p3", "w3"): 180,
  ("w1", "r1"): 160,
  ("w1", "r2"): 190,
  ("w1", "r3"): 110,
  ("w1", "r4"): 180,
  ("w1", "r5"): 150,
  ("w2", "r1"): 170,
  ("w2", "r2"): 190,
  ("w2", "r3"): 150,
  ("w2", "r4"): 140,
  ("w2", "r5"): 120,
  ("w3", "r1"): 140,
  ('w3', "r2"): 160,
  ("w3", "r3"): 180,
  ("w3", "r4"): 120,
  ("w3", "r5"): 100})

c = {
  ("p1", "w1"): 25,
  ("p1", "w2"): 85,
  ("p1", "w3"): 25,
  ("p2", "w1"): 50,
  ("p2", "w2"): 35,
  ("p2", "w3"): 95,
  ('p3', "w1"): 50,
  ("p3", "w2"): 40,
  ("p3", "w3"): 55,
  ("w1", "r1"): 75,
  ("w1", "r2"): 50,
  ("w1", "r3"): 60,
  ("w1", "r4"): 75,
  ("w1", "r5"): 30,
  ("w2", "r1"): 85,
  ("w2", "r2"): 15,
  ("w2", "r3"): 85,
  ("w2", "r4"): 85,
  ("w2", "r5"): 90,
  ("w3", "r1"): 90,
  ('w3', "r2"): 85,
  ("w3", "r3"): 35,
  ("w3", "r4"): 35,
  ("w3", "r5"): 95}


e = {
  ("A"): 1,
  ("B"): 3}


d = {

  ("w1","A"): 0,
  ("w2","A"): 0,
  ("w3","A"): 0,
  ("w1","B"): 0,
  ("w2","B"): 0,
  ("w3","B"): 0,
  ("r1","A"): 175,
  ("r2","A"): 120,
  ("r3","A"): 140,
  ("r4","A"): 100,
  ("r5","A"): 160,
  ("r1","B"): 100,
  ("r2","B"): 150,
  ("r3","B"): 110,
  ("r4","B"): 300,
  ("r5","B"): 230
}

phi = {
    ("p1", "A"): 0.5,
    ("p1", "B"): 0.1,
    ("p2", "A"): 0.5,
    ("p2", "B"): 0.1,
    ("p3", "A"): 0.2,
```

```
    ("p3", "B"): 0.4
}

q = {
    ("p1"): 90,
    ("p2"): 120,
    ("p3"): 80,
}

o = 48
m = 10000

model = Model()
model.__len__ = 1
x = model.addVars(A, L, vtype=GRB.INTEGER, name='x')
z = model.addVars(R, L, vtype=GRB.INTEGER, name="z")

obj1 = sum(c[i, j]*e[l]*x[i, j, l] for i, j in A for l in L) + sum(m*z[i, l] for i in R for l in L)


model.addConstrs((((sum(x[i, j, l]/phi[i, l] for l in L for j in N if (i, j) in A)) <= o*q[i]) for i in P), "c1")

model.addConstrs((x.sum(i, '*', l)-x.sum('*', i, l) == -d[i, l]+z[i, l] for l in L for i in R), "c2")

model.addConstrs((x.sum(i, '*', l)-x.sum('*', i, l) == -d[i, l] for l in L for i in W), "c3")

model.addConstrs((((sum(e[l]*x[i, j, l] for l in L)) <= u[i, j]) for (i, j) in A), "c5")

model.setObjective(obj1, GRB.MINIMIZE)

model.optimize()
model.printAttr('X')

Optimal solution found (tolerance 1.00e-04)
Best objective 6.601150000000e+06, best bound 6.601150000000e+06, gap 0.0000%

    Variable            X
-------------------------
  x[p1,w1,B]           40
  x[p1,w2,A]          150
  x[p1,w3,A]           77
  x[p1,w3,B]           31
  x[p2,w1,A]           81
  x[p2,w1,B]           23
  x[p2,w2,A]           67
  x[p2,w2,B]           31
  x[p2,w3,B]           60
  x[p3,w1,A]          150
  x[p3,w2,A]          170
  x[p3,w3,B]           60
  x[w1,r2,A]            1
  x[w1,r2,B]           63
  x[w1,r3,A]           80
  x[w1,r5,A]          150
  x[w2,r1,A]          169
  x[w2,r2,A]          118
  x[w2,r2,B]           24
  x[w2,r3,B]            7
  x[w2,r4,A]          100
  x[w3,r1,A]            6
  x[w3,r1,B]           18
  x[w3,r2,A]            1
  x[w3,r2,B]           53
  x[w3,r3,A]           60
  x[w3,r3,B]           40
  x[w3,r4,B]           40
  x[w3,r5,A]           10
     z[r1,B]           82
     z[r2,B]           10
     z[r3,B]           63
     z[r4,B]          260
     z[r5,B]          230
```

Part 3 (10 points) (Bonus for undergraduates/ mandatory for graduates) Capacitated nodes

There is also limitation for handling material by warehouses. Meaning that only certain weights of products can be handled by each warehouse. Table 10 show the maximum weight that can be handled by each warehouse.

Table 10: Maximum weight of product can be handled by the warehouses (in pounds)

Warehouse 1	Warehouse 2	Warehouse 3
660	700	1000

a (5 points): Formulate the necessary constraint required to address the material handling capacity of each warehouse.

t_i: Warehouse material handling capacity $i \in W$

$$\sum_{j:(i,j)\in A} \sum_{l\in L} e_l x_{ijl} \leq t_i \qquad \forall i \in W$$

b (5 points): Solve the problem using Excel or Gurobi to find the optimal transportation strategy (number of each product shipped from each plant to each warehouse, and from each warehouse to each retailer) with respect to total cost minimization.

```
model = Model()
model.__len__ = 1
x = model.addVars(A, L, vtype=GRB.INTEGER, name='x')
z = model.addVars(R, L, vtype=GRB.INTEGER, name="z")

obj1 = sum(c[i, j]*e[l]*x[i, j, l] for i, j in A for l in L) + sum(m*z[i, l] for i in R for l in L)

model.addConstrs((((sum(x[i, j, l]/phi[i, l] for l in L for j in N if (i, j) in A)) <= o*q[i]) for i in P), "c1")

model.addConstrs((x.sum(i, '*', l)-x.sum('*', i, l) == -d[i, l]+z[i, l] for l in L for i in R), "c2")

model.addConstrs((x.sum(i, '*', l)-x.sum('*', i, l) == -d[i, l] for l in L for i in W), "c3")

model.addConstrs((((sum(e[l]*x[i, j, l] for l in L)) <= u[i, j]) for (i, j) in A), "c5")

model.addConstrs(((sum(e[l]*x[i, j, l] for l in L for j in R if (i, j) in A)) <= t[i] for i in W), "c6")

model.setObjective(obj1, GRB.MINIMIZE)

model.optimize()
model.printAttr('X')
model.write('jigigoo.lp')

Optimal solution found (tolerance 1.00e-04)
Best objective 6.601150000000e+06, best bound 6.601150000000e+06, gap 0.0000%

    Variable            X
-------------------------
  x[p1,w1,B]           40
  x[p1,w2,A]          150
  x[p1,w3,A]           77
  x[p1,w3,B]           31
  x[p2,w1,A]          150
  x[p2,w2,A]           67
  x[p2,w2,B]           31
  x[p2,w3,B]           60
  x[p3,w1,A]           81
  x[p3,w1,B]           23
  x[p3,w2,A]          170
  x[p3,w3,B]           60
  x[w1,r2,A]            1
  x[w1,r2,B]           63
  x[w1,r3,A]           80
  x[w1,r5,A]          150
  x[w2,r1,A]          169
  x[w2,r2,A]          118
  x[w2,r2,B]           24
  x[w2,r3,B]            7
  x[w2,r4,A]          100
  x[w3,r1,A]            6
  x[w3,r1,B]           18
  x[w3,r2,A]            1
  x[w3,r2,B]           53
  x[w3,r3,A]           60
  x[w3,r3,B]           40
```

```
 x[w3,r4,B]            40
 x[w3,r5,A]            10
    z[r1,B]            82
    z[r2,B]            10
    z[r3,B]            63
    z[r4,B]           260
    z[r5,B]           230
```