Problem 1. Single-echelon Facility Location problem

New York city has 10 trash districts and is trying to determine which of the districts should be a site for dumping trash. It costs USD 1000 to haul one ton of trash one mile. The location of each district, the number of tons of trash produced per year by the district, the annual fixed cost (in millions of dollars) of running a dumping site, and the variable cost (per ton) of processing a ton of trash at a site are shown in the table

District	Coordinates		Tons	Cost	
	x	y		Fixed	Variable
1	4	2	49	2	310
2	2	5	874	1	40
3	10	8	555	1	51
4	2	8	352	1	341
5	5	3	381	3	131
6	4	5	428	2	182
7	10	5	985	1	20
8	5	1	105	2	40
9	5	8	258	4	177
10	1	7	210	2	75

Each dump site can handle at most 1 500 tons of trash. Each district must send all its trash to a single site. Determine how to locate the dump sites in order to minimize total cost per year.

1) Mathematical model

Mathematical model	**AMPL names:**
Formulation:	
1) $min \sum_{i \epsilon I} f_i y_i + \sum_{(i,j) \epsilon IJ} c_{ij} d_j x_{ij} + \sum_{(i,j) \epsilon IJ} x_{ij} d_j z_i$	Total_Cost
st	
2) $\sum_{(i,j) \in IJ} X_{ij} = 1 \ \ \forall i \in I$	Single_Site {j in J}
3) $\sum_{(i,j) \in IJ} d_j X_{ij} = 1 \ \ \forall i \in I$	Capacity {i in I}
4) $y_i \in \{0, 1\} \ \ \forall i \in I$	Location {J}
5) $X_{ij} \in \{0, 1\} \ \ \forall ij \in IJ$	Route {I, J}
Notation:	
Sets:	
J – locations to be a site for dumping trash	J
I – districts	I
Parameters:	
f_i – cost of running a dumping site $i \in I$	fc {J}
v_i – cost of processing a ton of trash at a site $i \in I$	vc {J}
s_{ij} – distance between districts $i \in I$ and $j \in J$	s {i in I, j in J}
w_i – amount of waste produced by district $i \in I$	w {I}
r – maximum capacity of a dump	capacity
x_j – x coordinate of district $j \in J$	x {j in J}
y_j – y coordinate of district $j \in J$	j {j in J}
Variables:	
X_j – 1 if the location j is used to locate the site	Location {J}

Z_{ij} – 1 if there's a route from j to i Route {I, J}

Description

The objective function (1) expresses the total cost of setting up dumping sites, running them and transporting trash to them. Constraint (2) ensures that there's a limit on capacity. Constraint (2) ensures that each district has only one dumping site. Constraints (4) and (5) ensure that variables are binary.

Problem size

The resulting model has following dimensions:

2 variables

5 constraints

2) AMPL code

```
File task1.dat:
set I := 1 2 3 4 5 6 7 8 9 10;
set J := 1 2 3 4 5 6 7 8 9 10;

param haul:= 1000;
param capacity:= 1500;

param x :=
     1 4
     2 2
     3 10
     4 2
     5 5
     6 4
     7 10
     8 5
     9 5
     10 1;
param y :=
     1 2
     2 5
     3 8
     4 8
     5 3
     6 5
     7 5
     8 1
     9 8
     10 7;
param w :=
     1 49
     2 874
     3 555
     4 352
     5 381
```

```
          6 428
          7 985
          8 105
          9 258
          10 210;
     param fc :=
          1 2000000
          2 1000000
          3 1000000
          4 1000000
          5 3000000
          6 2000000
          7 1000000
          8 2000000
          9 4000000
          10 2000000;
     param vc :=
          1 310
          2 40
          3 51
          4 341
          5 131
          6 182
          7 20
          8 40
          9 177
          10 75;

     File task1.mod:
     set I;
     set J;

     param haul; # cost of hauling one ton of trash one mile
     param capacity;
     param x {j in J};
     param y {j in J};
     param w {I}; # amount of waste
     param fc {J}; # fixed costs
     param vc {J}; # variable costs
     param s {i in I, j in J} := sqrt((x[j] - x[i])^2 + (y[j] - y[i])^2); # distance

     var Location {J} binary; # if district should be a site for dumping trash
     var Route {I,J} binary;
```

```
minimize Total_Cost:
       sum{j in J} Location[j] * fc[j]
     + sum{i in I, j in J} (haul * s[i,j]) * Route[i,j]  * w[i]
     + sum{i in I, j in J} vc[j] * Route[i,j] * w[i];

subject to Capacity {j in J}:
     sum{i in I} w[i] * Route[i,j] <= Location[j] * capacity;
subject to Single_Site {i in I}:
     sum{j in J} Route[i,j] = 1;
```

File task1.run

```
option solver cplex;
option cplex_options 'sensitivity';
model task1.mod;
data task1.dat;
solve;
# option omit_zero_rows 1;
display Total_Cost > task1.sol;
display Location > task1.sol;
display Route > task1.sol;
```

File task1.sol

```
Total_Cost = 9040700

Location [*] :=
 1  0
 2  1
 3  1
 4  1
 5  0
 6  1
 7  1
 8  0
 9  0
10  0
;

Route [*,*]
:    1   2   3   4   5   6   7   8   9  10    :=
1    0   0   0   0   0   1   0   0   0   0
2    0   1   0   0   0   0   0   0   0   0
3    0   0   1   0   0   0   0   0   0   0
4    0   0   0   1   0   0   0   0   0   0
5    0   0   0   0   0   1   0   0   0   0
```

```
6     0    0    0    0    0    1    0    0    0    0
7     0    0    0    0    0    0    1    0    0    0
8     0    0    0    0    0    1    0    0    0    0
9     0    0    0    1    0    0    0    0    0    0
10    0    0    0    1    0    0    0    0    0    0
;
```

3) Solution

To minimize total cost, dumps should be located at districts 2, 3, 4, 6, 7. Trash from districts 1, 5 and 8 should be hauled to district 6. Trash from districts 9 and 10 should be processed at district 4.

Total cost in this case will be 9040700.

Problem 2. Two-echelon Multi-commodity Facility Location Problem

A company sells two different products, manufactured at two factories, to three customers. Production capacities for the factories together with customers' demands for both products in units are:

Product	Demand			Capacity	
	Customer 1	Customer 2	Customer 3	Factory 1	Factory 2
1	300	400	500	600	700
2	200	300	600	400	800

The company wants to establish one or more depots in at most 6 different locations that have been identified as potential sites for such facilities. Each depot is planned to have a total capacity of 1400 units. The fixed costs incurred by establishing the depots are 3000, 3100, 2800, 2900, 3200 and 3000 for the corresponding 6 sites. The unit distribution costs for each of the products from factories to depots are given below.

	Depot											
	Product 1						Product 2					
Factory	1	2	3	4	5	6	1	2	3	4	5	6
1	5	7	3	8	6	4	7	9	4	9	9	6
2	6	4	5	6	7	2	9	7	7	10	11	5

The unit distribution costs from depots to customers for each of the products are:

	Customer					
	Product 1			Product 2		
Depot	1	2	3	1	2	3
1	10	12	9	13	15	11
2	8	11	13	10	15	17
3	6	14	7	9	16	11
4	5	11	15	8	12	18
5	10	17	12	14	21	15
6	8	16	16	12	20	21

a) Find the optimal distribution pattern and the sites where depots should be located.

1) Mathematical model

Mathematical model	**AMPL names:**
Formulation:	
1) $min \sum_{j\epsilon J} f_j y_j + \sum_{(i,j)\epsilon IJ} \sum_{p\epsilon P} b_{ijp} Z_{ijp} + \sum_{(j,k)\epsilon JK} c_{jkp} X_{jkp}$	Total_Cost
st	
2) $\sum_{(i,j)\epsilon IJ} \sum_{p\epsilon P} Z_{ijp} \leq s_{ip},\ \forall\, i \in I$	Capacity {i in I, p in PROD}
3) $\sum_{(j,k)\epsilon JK} \sum_{p\epsilon P} X_{jkp} \leq q_j y_j,\ \forall\, j \in J$	Depot_Capacity {j in J}
4) $\sum_{(i,j)\epsilon IJ} Z_{ijp} = \sum_{(j,k)\epsilon JK} X_{jkp}\ ,\ \forall\, j \in J\ \ \forall\, p \in P$	Amount {j in J, p in PROD}
5) $\sum_{(j,k)\epsilon JK} X_{jkp} = d_{kp}\ ,\ \forall\, k \in K\ \ \forall\, p \in P$	Demand {k in K, p in PROD}
6) $y_j \in \{0,1\},\ \forall\, j \in J$	Location {i in J}
7) $Z_{ijp} \geq 0, \forall\, (i,j) \in IJ$	IJ_flow {IJ, p in PROD}
8) $X_{jkp} \geq 0, \forall\, (j,k) \in JK\ \ \forall\, p \in P$	JK_flow {JK, p in PROD}
Notation:	
Sets:	
I – set of factories	I
J – set of potential sites for depots	J
K – set of customers	K
P – set of products	PROD
IJ – set of links between the factories and depots	IJ
JK – set of links between the depots and customers	JK
Parameters:	
d_{kp} – demand of customer k for product p	demand {K, PROD}
s_{ip} – capacity of the factory i for product p	capacity {I, PROD}
q – depots capacity	dc
f_i – cost of depot location at site j	fc {J}
b_{ijp} – unit transportation cost from factory i to depot j of product p	IJ_cost {IJ, p in PROD}
c_{jkp} – unit transportation cost from depot j to customer k of product p	JK_cost {JK, p in PROD}
Variables:	
X_{jkp} – flow of product p from depot j to customer k	IJ_flow {IJ, p in PROD}
Z_{ijp} - flow of product p from factory i to depot j	JK_flow {JK, p in PROD}
y_j - if depot is located at the site j, 0 otherwise	Location {j in J} binary

Description

The objective function (1) expresses the total cost of setting up and using depots. Constraint (2) ensures that there's a limit on capacity. Constraint (2) is a capacity constraint. Constraint (3) ensures that the depot capacity isn't exceeded. Constraint (4) is flow balance constraint. Constraint (5) is a demand constraint. Constraint (6) shows that the variable of choice of depots is binary. The constraints (7, 8) restrict the possibility to send negative amount of goods.

Problem size

The resulting model has following dimensions:

66 variables

28 constraints

2) AMPL code

```
File task2.dat:
set I:= F1 F2; #set of factories
```

```
set J:= D1 D2 D3 D4 D5 D6; #set of depots
set K:= C1 C2 C3; #set of customers
set PROD:= P1 P2; #set of products

param fc:= D1 3000 D2 3100 D3 2800 D4 2900 D5 3200 D6 3000;
param dc:= 1400; #depots' capacity

param demand: P1 P2 :=
C1 300 200
C2 400 300
C3 500 600;

param capacity: P1 P2:=
F1 600 400
F2 700 800;

param IJ_cost:=
[*,*,P1]: D1 D2 D3 D4 D5 D6:=
F1 5 7 3 8 6 4
F2 6 4 5 6 7 2

[*,*,P2]: D1 D2 D3 D4 D5 D6:=
F1 7 9 4 9 9 6
F2 9 7 7 10 11 5;

param JK_cost:=
[*,*,P1]: C1 C2 C3:=
D1 10 12 9
D2 8 11 13
D3 6 14 7
D4 5 11 15
D5 10 17 12
D6 8 16 16

[*,*,P2]: C1 C2 C3:=
D1 13 15 11
D2 10 15 17
D3 9 16 11
D4 8 12 18
D5 14 21 15
D6 12 20 21;
```

File task2.mod:

```
set I;
set J;
```

```
set K;
set PROD;

set IJ:={i in I, j in J};
set JK:= {j in J, k in K};

param demand {K, PROD};
param capacity {I,PROD};

param IJ_cost {IJ, p in PROD}>=0;
param JK_cost {JK, p in PROD}>=0;
param fc {J};
param dc;

var IJ_flow {IJ, p in PROD}>=0;
var JK_flow {JK, p in PROD}>=0;
var Location {i in J} binary;

minimize Total_Cost:
     sum{j in J} Location[j]*fc[j]
     + sum{(i,j) in IJ, p in PROD}IJ_flow[i,j,p]*IJ_cost[i,j,p]
     + sum{(j,k) in JK, p in PROD}JK_flow[j,k,p]*JK_cost[j,k,p];
subject to Capacity{i in I, p in PROD}:
     sum{(i,j) in IJ}IJ_flow[i,j,p]<= capacity[i,p];
subject to Depot_Capacity{j in J}:
     sum{(j,k) in JK, p in PROD}JK_flow[j,k,p]<=Location[j]*dc;
subject to Amount{j in J, p in PROD}:
     sum{(i,j) in IJ}IJ_flow[i,j,p] = sum{(j,k) in JK}JK_flow[j,k,p];
subject to Demand{k in K, p in PROD}:
     sum{(j,k) in JK}JK_flow[j,k,p] = demand[k,p];
```

File task2.run:

```
option solver cplex;
model task2b.mod;
data task2b.dat;
solve;
option omit_zero_rows 1;
display Total_Cost > task2.sol;
display Location > task2.sol;
display IJ_flow > task2.sol;
display JK_flow > task2.sol;
```

File task2.sol:

```
Total_Cost = 39600

Location [*] :=
D2  1
D3  1
;

IJ_flow :=
F1 D3 P1   600
F1 D3 P2   400
F2 D2 P1   400
F2 D2 P2   500
F2 D3 P1   200
F2 D3 P2   200
;

JK_flow :=
D2 C1 P2   200
D2 C2 P1   400
D2 C2 P2   300
D3 C1 P1   300
D3 C3 P1   500
D3 C3 P2   600
;
```

3) Solution

Depots should be located at D2 and D3.
Total cost will then be 39600$.

b) Geographically 6 potential sites are located in 2 regions, namely sites 1,2,3,4 - in region R1, and sites 5 and 6 - in region R2. The company wants at least one depot in each region and not more than three depots in total. Modify model from a) and find the resulting distribution pattern.

1) Mathematical model

Mathematical model	**AMPL names:**
Formulation:	
1) $min \sum_{j \epsilon J} f_j y_j + \sum_{(i,j)\epsilon IJ} \sum_{p \epsilon P} b_{ijp} Z_{ijp} + \sum_{(j,k)\epsilon JK} c_{jkp} X_{jkp}$	Total_Cost
st	
2) $\sum_{(i,j)\epsilon IJ} \sum_{p \epsilon P} Z_{ijp} \leq s_{ip},\ \forall\, i \in I$	Capacity {i in I, p in PROD}
3) $\sum_{(j,k)\epsilon JK} \sum_{p \epsilon P} X_{jkp} \leq q_j y_j,\ \forall\, j \in J$	Depot_Capacity {j in RR}
4) $\sum_{(i,j)\epsilon IJ} Z_{ijp} = \sum_{(j,k)\epsilon JK} X_{jkp},\ \forall\, j \in J\ \ \forall\, p \in P$	Amount {j in RR, p in PROD}
5) $\sum_{(j,k)\epsilon JK} X_{jkp} = d_{kp},\ \forall\, k \in K\ \ \forall\, p \in P$	Demand {k in K, p in PROD}
6) $y_j \in \{0,1\},\ \forall\, j \in J$	Location {i in RR}

7) $Z_{ijp} \geq 0, \forall\ (i,j) \in IJ$	IJ_flow {IJ, p in PROD}
8) $X_{jkp} \geq 0, \forall\ (j,k) \in JK\ \ \forall\ p \in P$	JK_flow {JK, p in PROD}
9) $\sum_{j\epsilon R1} y_j + \sum_{j\epsilon R2} y_j\ \leq l,\ \forall\ j \in R1\ \forall\ j \in R2$	DepotLimit
10) $\sum_{j\epsilon R1} y_j \geq 1,\ \forall\ j \in R1$	R1Limit
11) $\sum_{j\epsilon R2} y_j \geq 1,\ \forall\ j \in R2$	R2Limit

Notation:

Sets:

I – set of factories	I
J – set of potential sites for depots	J
K – set of customers	K
P – set of products	PROD
IJ – set of links between the factories and depots	IJ
JK – set of links between the depots and customers	JK
R1 – set of potential sites for depots in region 1	R1
R2 – set of potential sites for depots in region 2	R2

Parameters:

d_{kp} – demand of customer k for product p	demand {K, PROD}
s_{ip} – capacity of the factory i for product p	capacity {I, PROD}
q – depots capacity	dc
f_i – cost of depot location at site j	fc {J}
b_{ijp} – unit transportation cost from factory i to depot j of product p	IJ_cost {IJ, p in PROD}
c_{jkp} – unit transportation cost from depot j to customer k of product p	JK_cost {JK, p in PROD}
l – maximum number of depots	depot_limit

Variables:

X_{jkp} – flow of product p from depot j to customer k	IJ_flow {IJ, p in PROD}
Z_{ijp} - flow of product p from factory i to depot j	JK_flow {JK, p in PROD}
y_j - if depot is located at the site j, 0 otherwise	Location {j in J} binary

Description

The objective function (1) expresses the total cost of setting up and using depots. Constraint (2) is a capacity constraint. Constraint (3) ensures that the depot capacity isn't exceeded. Constraint (4) is flow balance constraint. Constraint (5) is a demand constraint. Constraint (6) shows that the variable of choice of depots is binary. The constraints (7, 8) restrict the possibility to send negative amount of goods. Constraints (9, 10, 11) are depot limit constraints.

Problem size

The resulting model has following dimensions:

66 variables
31 constraints

2) AMPL code

```
File task2b.dat:
set R1:= D1 D2 D3 D4;
set R2:= D5 D6;
param depot_limit:=3;

File task2b.mod:
set I;
set K;
```

```
set PROD;
set R1;
set R2;
set RR:= R1 union R2;

set IJ:={i in I, j in RR};
set JK:= {j in RR, k in K};

param demand {K, PROD};
param capacity {I,PROD};
param depot_limit;
param IJ_cost {IJ, p in PROD}>=0;
param JK_cost {JK, p in PROD}>=0;
param fc {RR};
param dc;

var IJ_flow {IJ, p in PROD}>=0;
var JK_flow {JK, p in PROD}>=0;
var Location {RR} binary;

minimize Total_Cost:
     sum{j in RR} Location[j]*fc[j]
     + sum{(i,j) in IJ, p in PROD}IJ_flow[i,j,p]*IJ_cost[i,j,p]
     + sum{(j,k) in JK, p in PROD}JK_flow[j,k,p]*JK_cost[j,k,p];
subject to Capacity{i in I, p in PROD}:
     sum{(i,j) in IJ}IJ_flow[i,j,p]<= capacity[i,p];
subject to Depot_Capacity{j in RR}:
     sum{(j,k) in JK, p in PROD}JK_flow[j,k,p]<=Location[j]*dc;
subject to Amount{j in RR, p in PROD}:
     sum{(i,j) in IJ}IJ_flow[i,j,p] = sum{(j,k) in JK}JK_flow[j,k,p];
subject to Demand{k in K, p in PROD}:
     sum{(j,k) in JK}JK_flow[j,k,p] = demand[k,p];
subject to DepotLimit:
     sum{j in R1}Location[j]+sum{k in R2}Location[k]<=depot_limit;
subject to R1Lim:
     sum{j in R1}Location[j]>=1;
subject to R2Lim:
     sum{j in R2}Location[j]>=1;

File task2b.run:
option solver cplex;
model task2b.mod;
```

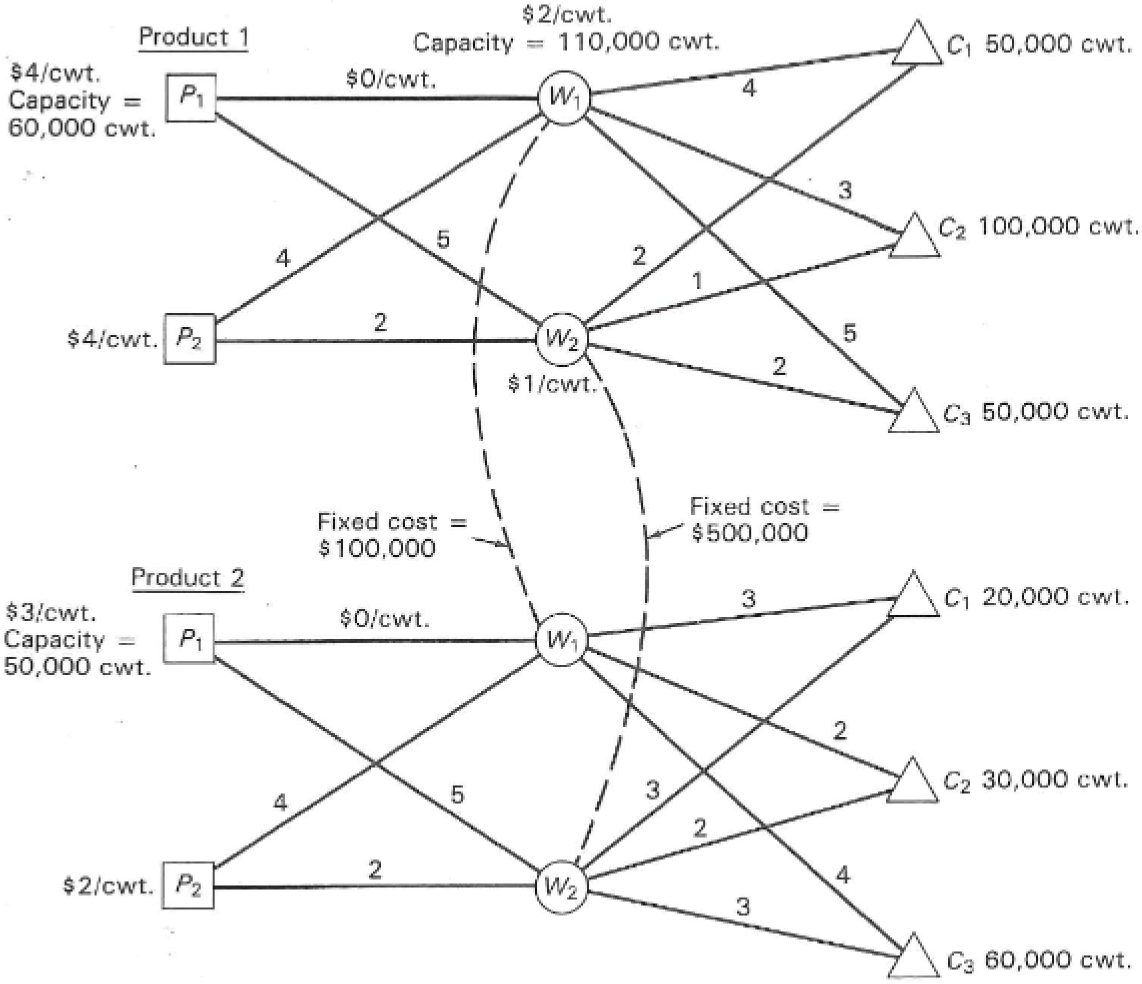
```
data task2b.dat;
solve;
option omit_zero_rows 1;
display Total_Cost > task2b.sol;
display Location > task2b.sol;
display IJ_flow > task2b.sol;
display JK_flow > task2b.sol;

File task2b.sol:
Total_Cost = 40900

Location [*] :=
D3  1
D6  1
;

IJ_flow :=
F1 D3 P1   500
F1 D3 P2   400
F2 D3 P2   500
F2 D6 P1   700
F2 D6 P2   200
;

JK_flow :=
D3 C2 P2   300
D3 C3 P1   500
D3 C3 P2   600
D6 C1 P1   300
D6 C1 P2   200
D6 C2 P1   400
;
```

3) Solution

Depots should be located at D3 and D6.
Total cost will then be 40900$.

Problem 3. Two-Echelon Network Design Problem

There are two products that are demanded by three customers, but a customer can be served out of only one of two warehouses. Warehouse 1 has a fixed cost of $100 000 per year if held open, a capacity of 110 000 cwt. per year, and a handling cost of $2/cwt. of throughput. Warehouse 2 has a fixed cost of $500 000, an unlimited capacity, and a handling cost of $1/cwt. Two plants can be used to serve the warehouses. Plant 1 has a production capacity of 60 000 cwt. for product 1 and 50 000 cwt. for product 2. Plant 2 has no capacity constraint for either product. Production and transportation costs in $ per cwt. and customer demands in cwt. for each product are shown in the picture below.



Find which warehouse(s) should be used, how customer demands should be assigned to them, and which warehouses and their throughput should be assigned to the plants.

1) Mathematical model

Mathematical model	**AMPL names:**
Formulation:	
1) $min \sum_{j \in J} f_j \, z_j + \sum_{(j,k) \in JK} \sum_{(p) \in P} h_j \, d_{kp} \, y_{jk} + \sum_{i \in I} \sum_{(j,k) \in JK} \sum_{p \in P} c_{ijkp} \, X_{ijkp}$	Total_Cost
st	
2) $\sum_{i \in I} X_{ijkp} \leq s_{ip}$, $\forall\, i \in I \; \forall\, p \in P$	Supply {i in I,p in P}
3) $\sum_{i \in I} X_{ijkp} = d_{kp} y_{jk}$, $\forall$ (j,k) $\in JK$ $\forall\, p \in P$	Demand {(j,k) in JK,p in P}
4) $\sum_{jk \in JK} y_{jk} = 1$, $\forall\, k \in K$	Single_Link {k in K}
5) $\sum_{jk \in JK} \sum_{p \in P} d_{kp} y_{jk} \leq q_j z_j$, $\forall j \in J$	Warehouse_Capacity {j in J}
6) $X_{ijkp} \geq 0$, $\forall i \in I, j \in J, k \in K, p \in P$	x {i in I,(j,k) in JK, p in P}
Notation:	
Sets:	
P – set of products	P

I – set of plants	I
J – set of warehouses	J
K – set of nodes	K
JK – potential links between warehouses and customers	JK
Parameters:	
d_{kp} – demand of customer $k \in K$ for product $p \in P$	demand {K, P}
s_{ip} – supply capacity for product p $\in P$ at plant $i \in I$	supply {I, P}
q_j – capacity of warehouse $j \in J$	q {J}
f_j – fixed cost of using warehouse $j \in J$	fc {J}
h_j – unit cost of throughput for warehouse at site $j \in J$	vc {J}
c_{ijkp} – unit cost of shipping product $p \in P$ from plant $i \in I$ through warehouse at site $j \in J$ to customer $k \in K$	cost {I, J, K, P}
Variables:	
x_{ijkp} – number of units to be shipped	x {i in I,(j,k) in JK, p in P}
y_{jk} – 1 if warehouse j is assigned to customer k	y {(j,k) in JK}
z_j – 1 warehouse is used	z {j in J}

Description

The objective function (1) expresses the total cost of using the network. Constraint (2) ensures that there's a limit on supply. Constraint (3) ensures that demand is satisfied. Constraint (4) ensures that customer is served by only one warehouse. Constraint (5) is warehouse capacity constraint. Constraint (6) ensures that the number of units transported is not negative.

Problem size

The resulting model has following dimensions:

32 variables
21 constraints

2) AMPL code

```
File task4.dat:
set P:= P1 P2;
set I:= I1 I2;
set J:= W1 W2;
set K:= K1 K2 K3;
set JK:= (W1,K1) (W1,K2) (W1,K3) (W2,K1) (W2,K2) (W2,K3);

param demand (tr):K1 K2 K3:=
P1 50000 100000 50000
P2 20000 30000 60000;

param supply (tr):I1 I2:=
P1 60000 10000000
P2 50000 10000000;

param q:= W1 110000 W2 10000000;
param fc:= W1 100000 W2 500000;
```

```
param vc:= W1 2 W2 1;

param cost [I1,W1,*,*]:
P1 P2:=
K1 4 8
K2 3 7
K3 5 9

[I1,W2,*,*]:
P1 P2:=
K1 7 4
K2 6 3
K3 7 4

[I2,W1,*,*]:
P1 P2:=
K1 3 7
K2 2 6
K3 4 8

[I2,W2,*,*]:
P1 P2:=
K1 8 5
K2 7 4
K3 8 5;
```

File task4.mod:

```
set P;
set I;
set J;
set K;
set JK within (J cross K);

param demand {K,P};
param supply {I,P};
param q {J} >= 0;
param fc {J} >= 0;
param vc {J} >= 0;
param cost {I,J,K,P} >= 0;

var z {j in J} binary;
var y {(j,k) in JK} binary;
var x {i in I,(j,k) in JK, p in P} >=0;

minimize Total_Cost:
```

```
		sum {j in J}fc[j]*z[j]
		+ sum {(j,k) in JK}sum {p in P}vc[j]*demand[k,p]*y[j,k]
		+ sum {i in I}sum {(j,k) in JK}sum {p in P}cost[i,j,k,p]*x[i,j,k,p];
	subject to Supply {i in I,p in P}:
		sum {(j,k) in JK} x[i,j,k,p] <= supply[i,p];
	subject to Demand {(j,k) in JK,p in P}:
		sum {i in I} x[i,j,k,p] = demand[k,p]*y[j,k];
	subject to Single_Link {k in K}:
		sum {(j,k) in JK} y[j,k] = 1;
	subject to Warehouse_Capacity {j in J}:
		sum {(j,k) in JK}sum {p in P} demand[k,p]*y[j,k] <= q[j]*z[j];
```

File task4.run:

```
option solver cplex;
model task4.mod;
data task4.dat;
solve;
option omit_zero_rows 1;
display Total_Cost > task4.sol;
display x > task4.sol;
display y > task4.sol;
display z > task4.sol;
```

File task4.sol:

```
Total_Cost = 2680000

x :=
I1 W2 K2 P1   60000
I1 W2 K3 P2   50000
I2 W1 K1 P1   50000
I2 W1 K1 P2   20000
I2 W2 K2 P1   40000
I2 W2 K2 P2   30000
I2 W2 K3 P1   50000
I2 W2 K3 P2   10000
;

y :=
W1 K1   1
W2 K2   1
W2 K3   1
;
```

```
z [*] :=
W1  1
W2  1
;
```

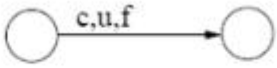
3) Solution

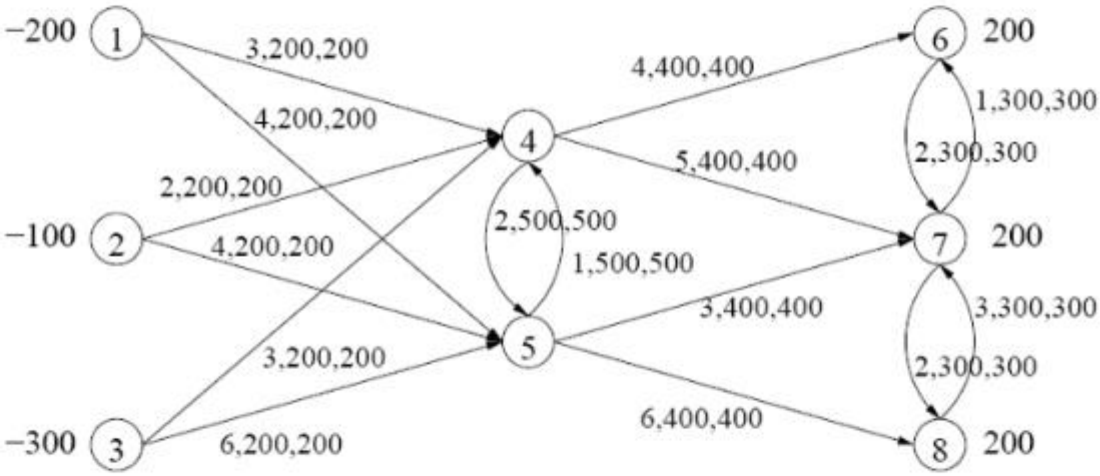
60000 cwt. of product 1 and 50000 cwt. of product 2 should be sent to customers 2 and 3 respectively from plant 1 through the warehouse 2. 50000 cwt. of product 1 and 20000 cwt. of product 2 should be sent to customer 1 from plant 2 through the warehouse 1. 40000 and 50000 cwt. of product 1 should be transported to the customers 2 and 3 respectively from plant 1 through the warehouse 2. 30000 and 10000 cwt. of product 2 should be sent to the customers 2 and 3 respectively from the plant 2 through the warehouse 2.

In this case total cost is 2680000$.

Problem 4. General Network Design Problem

Consider a network with three supply nodes 1, 2, 3, and three demand nodes 6,7 and 8. Numbers near these nodes show net flow to/from node = demand – supply. On each link you find value of unit shipment cost (c), arc capacity (u), and fixed cost to use arc (f).





Find which links to establish in a cost-efficient network, flows on the links, and total cost.

1) Mathematical model

Mathematical model	**AMPL names:**
Formulation:	
1) $min \sum_{(i,j)\in L} f_{ij}\ y_{ij} + \sum_{(i,j)\in L} c_{ij}\ X_{ij}$	Total_Cost
st	

2) $\sum_{(k,i)\in L} X_{ki} - \sum_{(i,j)\in L} X_{ij} = d_i - s_i$, $\forall\, i \in N$ Balance {i in NODES}
3) $l_{ij} \le X_{ij} \le u_{ij} y_{ij}$, $\forall\,(i,j) \in L$ Link_Capacity i,j) in LINKS}
4) $y_i \in \{0, 1\}$ $\forall i \in L$ y {(i,j) in LINKS} binary

Notation:

Sets:

N – set of nodes NODES
L – set of links LINKS

Parameters:

d_i – demand of node $i \in N$ demand {NODES}
s_i – supply of node $i \in N$ supply {NODES}
c_{ij} – unit shipping cost on link $(i, j) \in L$ cost {LINKS}
l_{ij} – lower limit on flow on link $(i, j) \in L$ ll {LINKS}
u_{ij} – upper limit on flow on link $(i, j) \in L$ ul {LINKS}
f_{ij} – cost of using the link $(i, j) \in L$ fc {LINKS}

Variables:

x_{ij} – number of units to be shipped x{(i,j) in LINKS}
y_{ij} – 1 if link is used, 0 otherwise y{(i,j) in LINKS}

Description

The objective function (1) expresses the total cost of setting up a network. Constraint (2) is a flow balance constraint. Constraint (3) is a link capacity constraint. Constraint (4) ensures that variables are binary.

Problem size

The resulting model has following dimensions:

32 variables
24 constraints

2) AMPL code

```
File task3.dat:
set NODES:= 1 2 3 4 5 6 7 8;
set LINKS:= (1,4) (1,5) (2,4) (2,5) (3,4) (3,5) (4,7) (4,6)
(4,5) (5,4) (5,7) (5,8) (6,7) (7,6) (7,8) (8,7);

param supply default 0:= 1 200 2 100 3 300;
param demand default 0:= 6 200 7 200 8 200;

param fc:=
1 4   200
1 5   200
2 4   200
2 5   200
3 4   200
3 5   200
4 7   400
4 6   400
4 5   500
5 4   500
5 7   400
```

```
5 8   400
6 7   300
7 6   300
7 8   300
8 7   300;

param cost:=
1 4   3
1 5   4
2 4   2
2 5   4
3 4   3
3 5   6
4 7   5
4 6   4
4 5   2
5 4   1
5 7   3
5 8   6
6 7   2
7 6   1
7 8   2
8 7   3;

param ll:=
1 4   0
1 5   0
2 4   0
2 5   0
3 4   0
3 5   0
4 7   0
4 6   0
4 5   0
5 4   0
5 7   0
5 8   0
6 7   0
7 6   0
7 8   0
8 7   0;

param ul:=
1 4   200
1 5   200
```

```
2 4   200
2 5   200
3 4   200
3 5   200
4 7   400
4 6   400
4 5   500
5 4   500
5 7   400
5 8   400
6 7   300
7 6   300
7 8   300
8 7   300;
```

File task3.mod:

```
set NODES;
set LINKS within (NODES cross NODES);
param supply {NODES};
param demand {NODES};
param cost {LINKS} >= 0;
param ll {LINKS} >= 0;
param ul {LINKS} >= 0;
param fc {LINKS} >= 0;
var y{(i,j) in LINKS} binary;
var X{(i,j) in LINKS}>=ll[i,j];

minimize Total_Cost:
     sum {(i,j) in LINKS}fc[i,j]*y[i,j] + sum {(i,j) in LINKS}cost[i,j]*X[i,j];

subject to Balance {i in NODES}:
     sum {(k,i) in LINKS} X[k,i] - sum {(i,j) in LINKS} X[i,j] = demand[i] - supply[i];

subject to Link_capacity {(i,j) in LINKS}:
     X[i,j]<= ul[i,j]*y[i,j];
```

File task3.run:

```
option solver cplex;
model task3.mod;
data task3.dat;
solve;
option omit_zero_rows 1;
display Total_Cost > task3.sol;
```

```
display X > task3.sol;
display y > task3.sol;

File task3.sol:
Total_Cost = 6700

X :=
1 5   200
2 5   100
3 4   200
3 5   100
4 6   200
5 7   400
7 8   200
;

y :=
1 5   1
2 5   1
3 4   1
3 5   1
4 6   1
5 7   1
7 8   1
;
```

3) Solution

Cost-efficient network flow should look like this: 200 units from node 1 to 5, 100 units from node 2 to 5, 200 – from 3 to 4, 100 – from 3 to 5, 200 – from 4 to 6, 400 – from 5 to 7, 200 – from 7 to 8.

In this case total cost will be 6700$.