Slide 1

	Per-Unit Shipping Cost to Customer					
Warehouse	1	2	3	4	5	Capacity
1	\$13	\$17	\$11	\$20	\$14	70
2	\$16	\$22	\$8	\$9	\$12	70
3	\$14	\$22	\$15	\$7	\$11	70
4	\$20	\$25	\$10	\$17	\$8	70
Demand	20	30	40	50	20	

Each warehouse cost \$1,000 to build.

Slide 2

MCNF Model

- $G = (N, A)$
  - $N = N_1 \cup N_2$ where
    - $N_1 = \{W_1, W_2, W_3, W_4\}$
    - $N_2 = \{C_1, C_2, C_3, C_4, C_5, D\}$
    - $b(i) = 70$ if $i \in N_1$
    - $b(C_1) = -20$, $b(C_2) = -30$, $b(C_3) = -40$, $b(C_4) = -50$, $b(C_5) = -20$
    - $b(D) = (20 + 30 + 40 + 50 + 20) - (4 \times 70) = -120$
  - $A = \{(i, j) : i \in N_1 \text{ and } j \in N_2\}$
    - c_{ij} = per-unit shipping cost if $j \in N_2 \setminus \{D\}$
    - $c_{iD} = 0$

Slide 3

Transportation Model

Ignoring the cost of constructing the warehouses gives the following solution:

```
CPLEX 6.6.0: optimal solution; objective 2200
x[W1,C1] = 20
x[W1,C2] = 30
x[W1,D] = 20
x[W2,C3] = 40
x[W2,D] = 30
x[W3,C4] = 50
x[W3,D] = 20
x[W4,C5] = 20
x[W4,D] = 50
```

Slide 4

Observe that all four warehouses are used in this solution. So, we need to add the construction cost of \$4,000 to the \$2,200 shipping cost given above to get a total cost of \$6,200. Since each warehouse can produce 70 units and the total demand is 160 units, we could possibly get by with only using three warehouses and save on \$1,000 in construction costs. Since the standard transportation problem does not capture the fixed costs (i.e., constructing the warehouse), we will have to take another approach.

Slide 5

Approach 1: Enumeration of Transportation Subproblems

We have already found the optimal four-warehouse solution. Since we must use at least three warehouses, we can solve three separate three-warehouse problems and compare the solutions to find the optimal overall solution. This is easy to do with an AMPL run file.

Slide 6

All 4 Warehouses

```
CPLEX 6.6.0: optimal solution; objective 2200
x[W1,C1] = 20
x[W1,C2] = 30
x[W1,D] = 20
x[W2,C3] = 40
x[W2,D] = 30
x[W3,C4] = 50
x[W3,D] = 20
x[W4,C5] = 20
x[W4,D] = 50
Construction cost = 4000
Shipping cost = 2200
Total cost = 6200
```


Slide 7

Warehouses 2, 3, & 4

```
CPLEX 6.6.0: optimal solution; objective 2370

x [*,*] (tr)
:    W1   W2   W3   W4    :=
C1    0    0   20    0
C2    0   30    0    0
C3    0   40    0    0
C4    0    0   50    0
C5    0    0    0   20
D    70    0    0   50
;

Construction cost = 3000
Shipping cost = 2370
Total cost = 5370
```

Slide 8

Warehouses 1, 2, & 3

```
CPLEX 6.6.0: optimal solution; objective 2280

x [*,*] (tr)
:    W1   W2   W3   W4    :=
C1   20    0    0    0
C2   30    0    0    0
C3    0    0    0   40
C4    0    0   50    0
C5    0    0    0   20
D    20   70   20   10
;

Construction cost = 3000
Shipping cost = 2280
Total cost = 5280
```

Slide 9

Warehouses 1, 2, & 4

```
CPLEX 6.6.0: optimal solution; objective 2340

x [*,*] (tr)
:    W1   W2   W3   W4    :=
C1   20    0    0    0
C2   30    0    0    0
C3    0   20    0   20
C4    0   50    0    0
C5    0    0    0   20
D    20    0   70   30
;

Construction cost = 3000
Shipping cost = 2340
Total cost = 5340
```

Slide 10

Warehouses 1, 2, & 3

```
CPLEX 6.6.0: optimal solution; objective 2260

x [*,*] (tr)
:    W1   W2   W3   W4    :=
C1   20    0    0    0
C2   30    0    0    0
C3    0   40    0    0
C4    0    0   50    0
C5    0    0   20    0
D    20   30    0   70
;

Construction cost = 3000
Shipping cost = 2260
Total cost = 5260
```


Slide 11

- optimal solution: build warehouses 1, 2, and 3.
- This approach is that it doesn't scale up well.
  - For example, if each warehouse could produce 160 units, then we would have to solve all 4 single-warehouse problems, all 6 two-warehouse problems in addition to the 4 three-warehouse problems and the four-warehouse problem.
  - It is easy to see that if we had two hundred potential warehouse locations and one thousand customers that this method would require solving an enormous number of transportation problems.

Slide 12

Approach 2: Integer Programming

The approach that is usually used in practice to solve this type of problem is to add *binary decision variables* to the transportation model. Let $y_i = 1$ if warehouse 1 is built and 0 otherwise. Let W be the set of potential warehouses and let C be the set of customers. With the binary variables, the objective function becomes:

$$\min \sum_{i \in W} \sum_{j \in C} c_{ij} x_{ij} + 1000 \sum_{i \in W} y_i.$$

Slide 13

To prevent shipments from warehouses that aren't build, we add the following set of constraints:

$$x_{ij} \leq 70 y_i \qquad \forall i \in W, j \in C.$$

Observe that when in any feasible solution setting $x_{ij} > 0$ for any j forces $y_i = 0$. Conversely, setting $y_i = 0$ forces $x_{ij} = 0$ for all j. An AMPL implementation of this model and the resulting solution are shown below.

Slide 14

```
# AMPL model for the warehouse problem
# By default, this model assumes that b[i] = 0, c[i,j] = 0,
# l[i,j] = 0 and u[i,j] = Infinity.


set NODES;              # nodes in the network
set W within NODES;     # set of warehouses
set C within NODES;     # set of customers

set ARCS within {NODES, NODES};     # arcs in the network

param b {NODES} default 0;          # supply/demand for node i
param c {ARCS} default 0;           # cost of one of flow on arc(i,j)
param l {ARCS} default 0;           # lower bound on flow on arc(i,j)
param u {ARCS} default Infinity;    # upper bound on flow on arc(i,j)
var x {ARCS};                       # flow on arc (i,j)
var y {W} binary;
```


Slide 15

```
minimize cost:
sum{(i,j) in ARCS} c[i,j] * x[i,j] +
sum{i in W} 1000 * y[i];

subject to flow_balance {i in NODES}:
sum{j in NODES: (i,j) in ARCS} x[i,j] -
sum{j in NODES: (j,i) in ARCS} x[j,i] = b[i];

subject to capacity {(i,j) in ARCS}:
l[i,j] <= x[i,j] <= u[i,j];

subject to build_warehouse {i in W, j in C}:
x[i,j] <= 70 * y[i];
```

Slide 16

```
# Run file for the warehouse problem
model warehouse_model.txt;
data  warehouse_data.txt;
let W := {'W1','W2','W3','W4'};
let C := {'C1','C2','C3','C4','C5'};
solve;
display x;
display y;
```

Slide 17

```
CPLEX 6.6.0: optimal integer solution;
objective 5260
93 MIP simplex iterations
16 branch-and-bound nodes
10 simplex iterations (4 in phase I)
x [*,*] (tr)
:    W1   W2   W3   W4    :=
C1   20    0    0    0
C2   30    0    0    0
C3    0   40    0    0
C4    0    0   50    0
C5    0    0   20    0
D    20   30    0   70
;
```

Slide 18

```
y [*] :=
W1  1
W2  1
W3  1
W4  0
;
```


Slide 19

As we would expect, the IP model gives the same solution as the first approach. In order to solve the IP, CPLEX actually solves a series of LP in which the y variables can take on any value in the range $[0, 1]$. CPLEX indicates this by reporting that there were 16 branch-and-bound nodes used to find the solution. Notice that if we don't require the y variables to be binary and try to solve the problem as an LP, we get a solution in which some of the arc flows are not integral. The problem is that once we add the extra set of constraints, we lose the network flow structure of the problem and the constraint matrix is no longer totally unimodular.

Slide 20

```
CPLEX 6.6.0:
CPLEX 6.6.0: optimal solution;
objective 3295.238095
15 simplex iterations (3 in phase I)
x [*,*] (tr)
:       W1       W2      W3          W4       :=
C1    3.33333    0   16.6667     0
C2    3.33333   10   16.6667     0
C3    3.33333   30    3.33333    3.33333
C4    0         30   16.6667     3.33333
C5    0          0   16.6667     3.33333
D    60          0    0         60
;
```

Slide 21

```
y [*] :=
W1  0.047619
W2  0.428571
W3  0.238095
W4  0.047619
;
```