Optimizing Shipping Routes for Rubber Duck Company

Charles Winston

February 20, 2018

1 The Problem

We want to minimize the total shipping costs for a company that sells rubber ducks with a supply chain system between several United States cities. The company has three warehouses in Santa Fe (SF), Tampa Bay (TB), and El Paso (EP). Each warehouse has a certain number of rubber ducks that must be shipped to various stores, represented in the following table.

SF	EP	TB
700	200	200

The company has 5 stores located across the US in Chicago (CH), LA, NY, Houston (HO), and Atlanta (AT). The demand in number of ducks for each of these cities is represented in the following table.

CH	LA	NY	HO	AT
200	200	250	300	150

In addition to the company's warehouses, HO and AT are hubs that in addition to their own demand, can also transfer materials to other store locations. The company's shipping costs along these routes are represented in the following table. Some routes are not available.

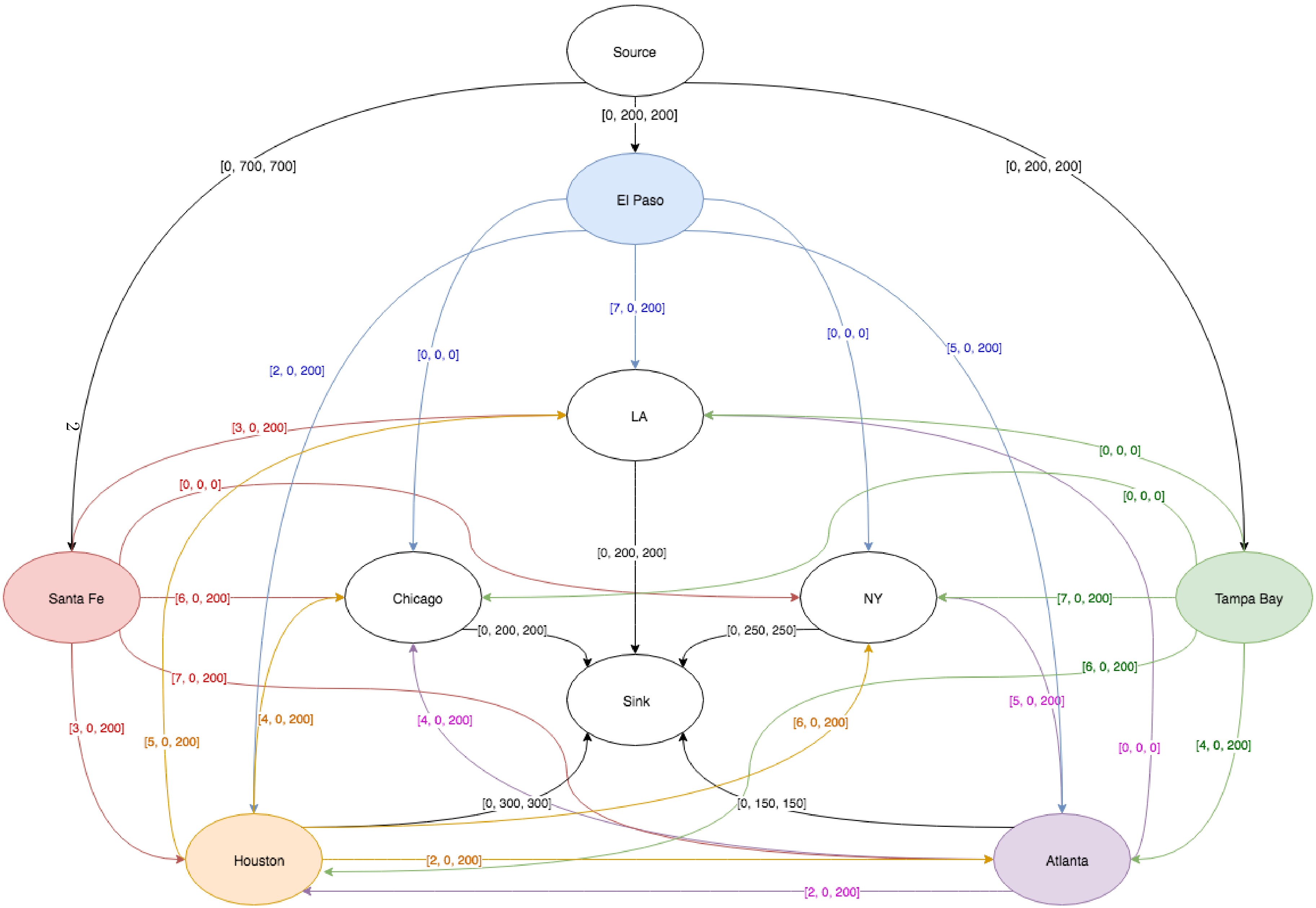
	CH	LA	NY	HO	AT
SF	\$6	\$3	-	\$3	\$7
EP	-	\$7	-	\$2	\$5
TB	-	-	\$7	\$6	\$4
HO	\$4	\$5	\$6	-	\$2
AT	\$4	-	\$5	\$2	-

Each route is restricted to shipping a maximum of 200 ducks.

Our goal is to come up with the optimal shipping plan for this company that will minimize shipping costs while at the same time meeting all customer demand. We can model this problem very well using a linear program.

2 Network Flow Model

We can set up a linear program for this problem by modeling the flow of rubber ducks across different cities with a directed graph. The cities will be represented by nodes in the graph, while the arcs between them will represent the variable number of rubber ducks moving along the shipping routes between the cities. We will also include a "source" node, which will represent the ducks initially moving to the warehouses, as well as a "sink" node, representing the ducks being sold to customers. Each arc will have a cost associated with moving a single rubber duck, and a lower and upper bound for how many rubber ducks can move across the arc. The diagram is presented on the next page, each arc labeled with a vector representing [cost, lowerbound, upperbound].


Source
[0, 700, 700]
[0, 200, 200]
[0, 200, 200]
El Paso
[7, 0, 200]
[0, 0, 0]
[0, 0, 0]
[2, 0, 200]
[5, 0, 200]
LA
[3, 0, 200]
[0, 0, 0]
[0, 0, 0]
[0, 0, 0]
[0, 200, 200]
Santa Fe
[6, 0, 200]
Chicago
NY
[7, 0, 200]
Tampa Bay
[0, 200, 200]
[0, 250, 250]
[7, 0, 200]
[6, 0, 200]
Sink
[3, 0, 200]
[5, 0, 200]
[4, 0, 200]
[4, 0, 200]
[6, 0, 200]
[5, 0, 200]
[0, 0, 0]
[4, 0, 200]
[0, 300, 300]
[0, 150, 150]
Houston
[2, 0, 200]
Atlanta
[2, 0, 200]

Any arc with a vector of all zeros represents an unavailable route. The arcs from the source all have zero cost and their upper bound equal to their lower bound, which is equal to the number of ducks that start at the warehouse the arc is connected to. This is because no ducks are actually moving from the source, but rather the arcs are there to represent the fixed starting number of ducks at each warehouse. Similarly, the arcs into the sink all have zero cost and their upper bound equal to their lower bound, which is equal to the demand at each city the arc is coming from. This is because no ducks are actually being shipped across these arcs, but these arcs represent the fixed amount of ducks that must be sold at each location.

All other arcs represent an actual possible route in the supply chain. Each of these arcs have an upper bound of 200 because that is the maximum amount of ducks that can be shipped across any route. These arcs also all have a lower bound of 0, because we can choose not to ship any ducks across this route, but we obviously cannot ship negative ducks. The cost for these arcs is simply the cost to ship one duck across the specified route, which is represented in the table in the introduction.

The last very important constraint is that for all non source or sink nodes, the amount of ducks going into it must equal the amount of ducks coming out. This constraint is described in more detail in the following section.

3 Linear Program

With this network flow model, we now have a good framework for a linear program. For each arc i:

- x_i will represent the number of ducks moving across the arc
- c_i will represent the cost to move a single duck across the arc
- l_i will represent the lower bound of x_i
- u_i will represent the upper bound for x_i

So from the [cost, lowerbound, upperbound] vector associated with each arc, we have the constants c_i, l_i, u_i for all i. We now label each arc with its index i. We will exclude arcs with all zeros, because they will not play any part in the linear program. In addition to the city abbreviations given above, we will also abbreviate the source node as SC and the sink node as SK.

1: SC → SF	6: SF → HO	11: TB → AT	16: HO → NY	21: AT → SK
2: SC → EP	7: SF → AT	12: TB → HO	17: HO → AT	22: HO → SK
3: SC → TB	8: EP → HO	13: TB → NY	18: AT → CH	23: NY → SK
4: SF → CH	9: EP → LA	14: HO → CH	19: AT → NY	24: LA → SK
5: SF → LA	10: EP → AT	15: HO → LA	20: AT → HO	25: CH → SK

One very important constraint in this linear program is related to the conservation of ducks moving through a node: the number of ducks going into a node minus the number of ducks moving out of the node must equal 0. The exceptions here are the source and sink nodes. We can represent this conservation with the following system of linear equations.

$$
\begin{aligned}
x_1 - x_4 - x_5 - x_6 - x_7 &= 0 && \text{SF conservation} \\
x_2 - x_8 - x_9 - x_{10} &= 0 && \text{EP conservation} \\
x_3 - x_{11} - x_{12} - x_{13} &= 0 && \text{TB conservation} \\
x_6 + x_8 + x_{12} + x_{20} - x_{14} - x_{15} - x_{16} - x_{17} - x_{22} &= 0 && \text{HO conservation} \\
x_7 + x_{10} + x_{11} + x_{17} - x_{18} - x_{19} - x_{20} - x_{21} &= 0 && \text{AT conservation} \\
x_4 + x_{14} + x_{18} - x_{25} &= 0 && \text{CH conservation} \\
x_5 + x_9 + x_{15} - x_{24} &= 0 && \text{LA conservation} \\
x_{13} + x_{16} + x_{19} - x_{23} &= 0 && \text{NY conservation}
\end{aligned}
$$

We can represent this system of linear equations with an 8×25 matrix A where each row represents the coefficients of the left side of each equation equal to 0. So we can define A as the following matrix.

$$A = \begin{bmatrix} 1&0&0&-1&-1&-1&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&-1&-1&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&-1&-1&-1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&1&0&0&0&1&0&-1&-1&-1&-1&0&0&1&0&-1&0&0&0 \\ 0&0&0&0&0&0&1&0&0&1&1&0&0&0&0&0&1&-1&-1&-1&-1&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&-1 \\ 0&0&0&0&1&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&-1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&1&0&0&0&-1&0&0 \end{bmatrix}$$

Now for a vector of 8 dimensions $\vec{b} = [0, \cdots, 0]$, the system of linear equations can be defined as $A\vec{x} = \vec{b}$.

We now have a cost vector $\vec{c} = [c_1, \cdots, c_{25}]$, a vector of unknowns $\vec{x} = [x_1, \cdots, x_{25}]$, a vector of lower bounds $\vec{l} = [l_1, \cdots, l_{25}]$, a vector of upper bounds $\vec{u} = [u_1, \cdots, u_{25}]$, a vector of 9 zeros $\vec{b}$, and a matrix A. We now have enough information to formally write out our linear program:

We want to minimize $\vec{c} \cdot \vec{x}$ subject to the following constraints.

$$A\vec{x} = \vec{b}$$
$$\vec{l} \leq \vec{x} \leq \vec{u}$$

4 Computing the Linear Program

We now have a linear program in a format easy to put into Matlab. Section 8 contains the code to run this linear program. The ordering of the indices for the vectors and matrices in the code is consistent to the ordering presented in section 3. The result of this linear program produced a minimum cost of \$5300 and the following for $\vec{x}$.

x_1: SC → SF = 700	x_6: SF → HO = 200	x_{11}: TB → AT = 0	x_{16}: HO → NY = 50	x_{21}: AT → SK = 150
x_2: SC → EP = 200	x_7: SF → AT = 100	x_{12}: TB → HO = 0	x_{17}: HO → AT = 50	x_{22}: HO → SK = 300
x_3: SC → TB = 200	x_8: EP → HO = 200	x_{13}: TB → NY = 200	x_{18}: AT → CH = 0	x_{23}: NY → SK = 250
x_4: SF → CH = 200	x_9: EP → LA = 0	x_{14}: HO → CH = 0	x_{19}: AT → NY = 0	x_{24}: LA → SK = 200
x_5: SF → LA = 200	x_{10}: EP → AT = 0	x_{15}: HO → LA = 0	x_{20}: AT → HO = 0	x_{25}: CH → SK = 200

We can ignore $x_1, x_2, x_3, x_{21}, x_{22}, x_{23}, x_{24}, x_{25}$ because these values are fixed and do not represent actual shipping routes. So the solution to the linear program are the values for $x_4, \cdots, x_{20}$, which are the optimal number of ducks to ship across each corresponding route to produce the minimum shipping cost of \$5300.

5 Problem Variants

Consider the following variant to the problem. The shipping workers in LA are unhappy and considering to strike. They demand that all shipping costs to LA be doubled, otherwise they strike and the maximum number of supplies that can be shipped on each route through LA is cut in half. We can model each of the two scenarios:

1. Avoid strike → all shipping costs to LA doubled.

2. Workers strike → maximum number of supplies shipped on each route through LA cut in half.

Scenario 1 produces the following changes to the model. In the cost vector $\vec{c}$, c_5 changes from \$3 to \$6, c_9 changes from \$7 to \$14, and c_{15} changes from \$5 to \$10. Computing this variant of the linear program results in a minimum cost of \$5900. It also produces the following values for the shipping routes, which are the same values as in the original problem.

x_4: SF $\rightarrow$ CH $= 200$	x_9: EP $\rightarrow$ LA $= 0$	x_{14}: HO $\rightarrow$ CH $= 0$	x_{19}: AT $\rightarrow$ NY $= 0$
x_5: SF $\rightarrow$ LA $= 200$	x_{10}: EP $\rightarrow$ AT $= 0$	x_{15}: HO $\rightarrow$ LA $= 0$	x_{20}: AT $\rightarrow$ HO $= 0$
x_6: SF $\rightarrow$ HO $= 200$	x_{11}: TB $\rightarrow$ AT $= 0$	x_{16}: HO $\rightarrow$ NY $= 50$	
x_7: SF $\rightarrow$ AT $= 100$	x_{12}: TB $\rightarrow$ HO $= 0$	x_{17}: HO $\rightarrow$ AT $= 50$	
x_8: EP $\rightarrow$ HO $= 200$	x_{13}: TB $\rightarrow$ NY $= 200$	x_{18}: AT $\rightarrow$ CH $= 0$	

Scenario 2 produces the following changes to the model. In the upper bound vector $\vec{u}$, u_5, u_9, and u_{15} each change from 200 to 100. Computing this variant of the linear program results in a minimum cost of \$6050. It also produces the following values for the shipping routes. Notice one route involving LA decreases from 200 to 100, and another route to make up for the loss increases from 0 to 100, so LA can meet its total demand of 200 rubber ducks.

x_4: SF $\rightarrow$ CH $= 200$	x_9: EP $\rightarrow$ LA $= 0$	x_{14}: HO $\rightarrow$ CH $= 0$	x_{19}: AT $\rightarrow$ NY $= 50$
x_5: SF $\rightarrow$ LA $= 100$	x_{10}: EP $\rightarrow$ AT $= 0$	x_{15}: HO $\rightarrow$ LA $= 100$	x_{20}: AT $\rightarrow$ HO $= 0$
x_6: SF $\rightarrow$ HO $= 200$	x_{11}: TB $\rightarrow$ AT $= 0$	x_{16}: HO $\rightarrow$ NY $= 0$	
x_7: SF $\rightarrow$ AT $= 200$	x_{12}: TB $\rightarrow$ HO $= 0$	x_{17}: HO $\rightarrow$ AT $= 0$	
x_8: EP $\rightarrow$ HO $= 200$	x_{13}: TB $\rightarrow$ NY $= 200$	x_{18}: AT $\rightarrow$ CH $= 0$	

We can see that Scenario 1 produces a lower minimum shipping cost, so it is better for the company to avoid the strike and appease the workers by raising the shipping costs to LA.

Now imagine the same scenario in Houston. More changes will occur in this case because Houston is a hub city with many routes coming in and going out of it. We can model the following scenarios:

1. Avoid strike $\rightarrow$ all shipping costs to and from Houston doubled.
2. Workers strike $\rightarrow$ maximum number of supplies shipped on each route through Houston cut in half.

Scenario 1 produces the following changes to the model. In the cost vector $\vec{c}$, c_6 changes from \$3 to \$6, c_8 changes from \$2 to \$4, c_{12} changes from \$6 to \$12, c_{14} changes from \$4 to \$8, c_{15} changes from \$5 to \$10, c_{16} changes from \$6 to \$12, c_{17} changes from \$2 to \$4, and c_{20} changes from \$2 to \$4. Computing this variant of the linear program results in a minimum cost of \$6250. It also produces the following values for the shipping routes. Notice that now, Houston is not used as a hub city at all and only routes coming into it ship any ducks.

x_4: SF $\rightarrow$ CH $= 200$	x_9: EP $\rightarrow$ LA $= 0$	x_{14}: HO $\rightarrow$ CH $= 0$	x_{19}: AT $\rightarrow$ NY $= 50$
x_5: SF $\rightarrow$ LA $= 200$	x_{10}: EP $\rightarrow$ AT $= 100$	x_{15}: HO $\rightarrow$ LA $= 0$	x_{20}: AT $\rightarrow$ HO $= 0$
x_6: SF $\rightarrow$ HO $= 200$	x_{11}: TB $\rightarrow$ AT $= 0$	x_{16}: HO $\rightarrow$ NY $= 0$	
x_7: SF $\rightarrow$ AT $= 100$	x_{12}: TB $\rightarrow$ HO $= 0$	x_{17}: HO $\rightarrow$ AT $= 0$	
x_8: EP $\rightarrow$ HO $= 100$	x_{13}: TB $\rightarrow$ NY $= 200$	x_{18}: AT $\rightarrow$ CH $= 0$	

Scenario 2 produces the following changes to the model. In the upper bound vector $\vec{u}$, u_6, u_8, u_{12}, u_{14}, u_{15}, u_{16}, u_{17}, and u_{20} each change from 200 to 100. Computing this variant of the linear program results in a minimum cost of \$6050. It also produces the following values for the shipping routes. The values are very similar to scenario one, except that x_6 decreases to 100 and x_{20} increases to 100, because of the constraint on the maximum amount of units shipped to Houston.

x_4: SF $\rightarrow$ CH $= 200$	x_9: EP $\rightarrow$ LA $= 0$	x_{14}: HO $\rightarrow$ CH $= 0$	x_{19}: AT $\rightarrow$ NY $= 50$
x_5: SF $\rightarrow$ LA $= 200$	x_{10}: EP $\rightarrow$ AT $= 100$	x_{15}: HO $\rightarrow$ LA $= 0$	x_{20}: AT $\rightarrow$ HO $= 100$
x_6: SF $\rightarrow$ HO $= 100$	x_{11}: TB $\rightarrow$ AT $= 0$	x_{16}: HO $\rightarrow$ NY $= 0$	
x_7: SF $\rightarrow$ AT $= 200$	x_{12}: TB $\rightarrow$ HO $= 0$	x_{17}: HO $\rightarrow$ AT $= 0$	
x_8: EP $\rightarrow$ HO $= 100$	x_{13}: TB $\rightarrow$ NY $= 200$	x_{18}: AT $\rightarrow$ CH $= 0$	

For this Houston variant, we can see that Scenario 2 produces a lower minimum shipping cost, so it would be better for the company to let the workers strike in this case.

It is clear that the result of these variants would be more drastic in Houston because it would result in a cost of $6050, rather than in LA which would result in a cost of $5900. In both variants, routes passing through the city with restrictions are avoided more than the others. So it makes sense that Houston would cause more problems because it is a hub city and there are more routes going through it, so more restrictions on this city would cause larger results across the entire model.

6 Maximizing Profit

The following table defines the profit made at each city for selling a single rubber duck.

SF	EP	TB	CH	LA	NY	HO	AT
-$8	-$5	-$10	$15	$25	$10	$10	$20

We can slightly change our original model in order to maximize profit. First, all values in our cost vector $\vec{c}$ must be made negative, because all the shipping costs represent a negative profit. Then, we change the costs from arcs connected to the source and the sink, because these arcs represent the profit values in the table given above. We have the following changes: $c_1 = -8, c_2 = -5, c_3 = -10, c_{21} = 10, c_{22} = 10, c_{23} = 25, c_{24} = 20, c_{25} = 15$. Now $\vec{c}$ represents the profit per duck, and we want to maximize $\vec{c} \cdot \vec{x}$ subject to the same constraints as the original linear program, $A\vec{x} = \vec{b}$ and $\vec{l} \leq \vec{x} \leq \vec{u}$.

When computing the linear program, we get a maximum profit of $2600. The resulting values for $\vec{x}$ are exactly the same as what were found in section 4. This is exactly what is expected because as long as we are required to meet total demand at all of the stores, then our revenue will always be constant no matter what shipping route we choose. Since profit is equal to revenue minus costs, then maximizing profit is the same as minimizing the costs. And so for both scenarios, maximizing total profit or minimizing total shipping costs, our resulting values for how many ducks to send across each route are the same.

7 Conclusions

In this report, we have shown that assuming there are no worker uprisings at any locations, we can get a minimum shipping cost of $5300 associated with a maximum profit of $2300 if we follow the optimal routes presented in section 4. We strongly advise the rubber duck company to follow the findings of this report so it can make as much money possible.

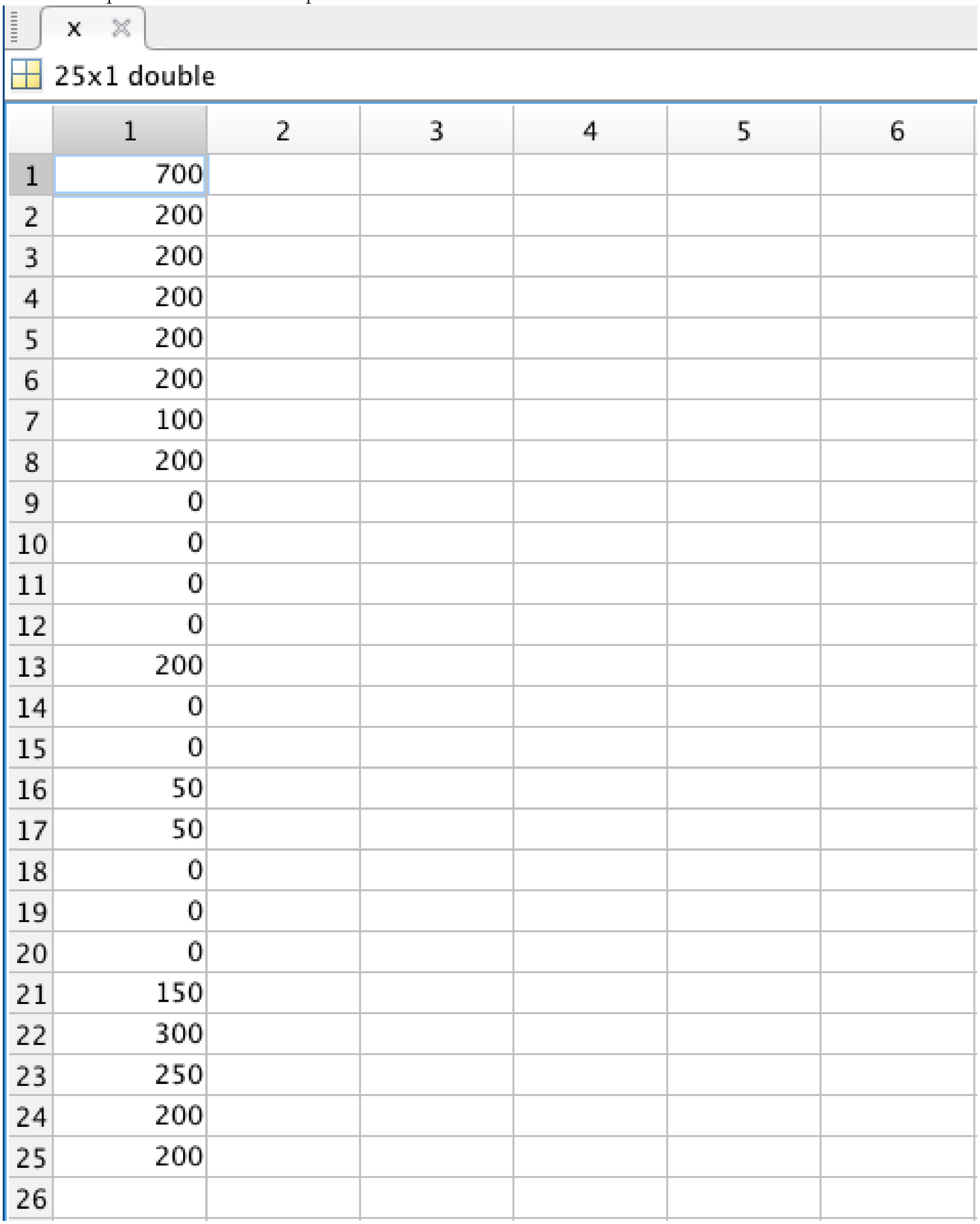
8 MatLab Code

The code is presented below.


```
% Linear program to minimize shipping costs for rubber duck company

cost = [0, 0, 0, 6, 3, 3, 7, 2, 7, 5, 4, 6, 7, 4, 5, 6, 2, 4, 5, 2, 0, 0, 0, 0, 0];
upper = [700, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200, 200,↙
200, 200, 200, 200, 150, 300, 250, 200, 200];
lower = [700, 200, 200, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 150, 300, 250,↙
200, 200];

beq = zeros(8, 1);
Aeq = zeros(8, 25);

% Santa Fe Conservation Constraints
Aeq(1, 1) = 1;
Aeq(1, 4) = -1;
Aeq(1, 5) = -1;
Aeq(1, 6) = -1;
Aeq(1, 7) = -1;

% El Paso Conservation Constraints
Aeq(2, 2) = 1;
Aeq(2, 8) = -1;
Aeq(2, 9) = -1;
Aeq(2, 10) = -1;

% Tamba Bay Conservation Constraints
Aeq(3, 3) = 1;
Aeq(3, 11) = -1;
Aeq(3, 12) = -1;
Aeq(3, 13) = -1;

% Houston Conservation Constraints
Aeq(4, 6) = 1;
Aeq(4, 8) = 1;
Aeq(4, 12) = 1;
Aeq(4, 14) = -1;
Aeq(4, 15) = -1;
Aeq(4, 16) = -1;
Aeq(4, 17) = -1;
Aeq(4, 20) = 1;
Aeq(4, 22) = -1;

% Atlanta Conservation Constraints
Aeq(5, 7) = 1;
Aeq(5, 10) = 1;
Aeq(5, 11) = 1;
Aeq(5, 17) = 1;
Aeq(5, 18) = -1;
Aeq(5, 19) = -1;
Aeq(5, 20) = -1;
Aeq(5, 21) = -1;

% Chicago Conservation Constraints
Aeq(6, 4) = 1;
Aeq(6, 14) = 1;
Aeq(6, 18) = 1;
Aeq(6, 25) = -1;

% LA Conservation Constraints
```

```
Aeq(7, 5) = 1;
Aeq(7, 9) = 1;
Aeq(7, 15) = 1;
Aeq(7, 24) = -1;

% NY Conservation Constraints
Aeq(8, 13) = 1;
Aeq(8, 16) = 1;
Aeq(8, 19) = 1;
Aeq(8, 23) = -1;


% Call linprog
options=optimset ('display', 'off');
x = linprog(cost, [], [], Aeq, beq, lower, upper, [], options);
min_cost = dot(cost, x)
```

The output of the function is presented below.



x

25x1 double

	1	2	3	4	5	6
1	700					
2	200					
3	200					
4	200					
5	200					
6	200					
7	100					
8	200					
9	0					
10	0					
11	0					
12	0					
13	200					
14	0					
15	0					
16	50					
17	50					
18	0					
19	0					
20	0					
21	150					
22	300					
23	250					
24	200					
25	200					
26						