‘

Spring | 11

Optimizing Product Transportation At Frito Lay

Javier González
Fernando G. Sada
Marcelo G. Sada

Southern Methodist University

Table of Contents

1. Management Summary

In order to satisfy the large Hispanic market in the United States, Frito Lay imports products manufactured in Mexico by Gamesa, a sister company to Frito Lay due to the fact that they are both owned by PepsiCo. Gamesa has plants all across Mexico, having the main ones located in Celaya, Vallejo, Obregon and Monterrey.

In the past, all products being imported into the U.S were produced in Obregon, a city in Northwestern Mexico. The main reason for this was Obregon's close proximity to California, which was Gamesa's main market in the U.S. As of today, the Hispanic population has dispersed all across the country, increasing Gamesa's market to a large portion of it. Gamesa's products are now being distributed all the way from California, to Texas, and onto the East Coast and Midwest. Imports have increased so much that in order to satisfy the demand, Gamesa has started producing all across Mexico for exports. It is important to mention that Obregon continues to be the main producer of the products being exported, followed by the plant in Monterrey.

Currently, all trucks that move products from distinct parts of Mexico into the U.S border pass through Obregon. Although this is convenient for deliveries to California, it might not be optimal for deliveries in other regions. Therefore, our project goal is to help Frito Lay reduce transportation costs of certain products (specified later) by finding the optimal routes to export/import from Mexico into the United States. This of course is based on the demand and production of the different products, which differ in great amounts.

To achieve the results we evaluated the current transportation method used by Frito Lay. We analyzed products, volumes, demands, routes and destinations to determine if it could be improved or not, and if so, how. We created a Multi-Commodity Fixed Charge Network Flow Model and used two computational packages called AMPL and CPLEX to find the optimal route. The results indicated that new routes should be considered for optimality and ultimately be able to reduce costs.

2. Background and Description of the Problem

Background

Gamesa, the largest cookie company in Mexico was established in 1921 by a group of businessmen. It was acquired by PepsiCo in 1990 and went from a domestic family business to a global company. Currently, Gamesa manufactures products for the Latin American and North American markets. Quaker U.S.A. and Frito Lay distribute most of Gamesa's export products.

Description of the Problem

In order to satisfy the consumer needs of the large (and incrementing) Hispanic population in the United States, PepsiCo, with two of its snacks subsidiaries, is taking advantages of the synergies created. These two companies are Frito Lay, the leading corn and potato chips manufacturer in the U.S, and Gamesa, the leading cookie and snack manufacturer in Mexico. The synergy here is obvious: Frito Lay currently imports some of Gamesa's products to be consumed in the U.S.

This all seems pretty simple, until you get into the details. Gamesa produces these snacks in many different plants across the country, the main ones being located in Vallejo, Celaya, Obregon and Monterrey.

In the past, all products being imported into the U.S were produced in Obregon, a city in Northwestern Mexico. The main reason for this was Obregon's close proximity to California, which was Gamesa's main market in the U.S. As of today, the Hispanic population has dispersed all across the country, increasing Gamesa's market to a large portion of the country. Gamesa's products are now being distributed all the way from California, to Texas, and onto the East Coast and Midwest. Imports have increased so much that in order to satisfy the demand, Gamesa has started producing all across Mexico for exports. It is important to mention that Obregon continues to be the main producer of the products being exported, followed by the plant in Monterrey.

The problem lies in optimality based on geography and capabilities of the plants. It is probably more efficient, and therefore less costly for products that are going to be consumed in Tennessee to be consolidated in Dallas rather than in Obregon.

Problem Situation

Speaking with Frito Lay and Gamesa employees about the scope for our project. They advised us to focus on a small set of products so that our analysis and model could be created within our time limit. For the purpose of the project, we focused on four products; Munchies crackers and Minifudge cookies both in 33 cents (1.22 oz) and 99 cents (33 oz) presentations.

These products we are focusing on are manufactured and packaged in the Gamesa plant of Vallejo. In our project we consider this city o Vallejo as our source**.** All products currently pass through a plant in Obregon, located in Northwest Mexico. There, products are consolidated and shipped to the warehouses in both East and West Coasts. The warehouses are located in: Pulaski (Tennessee), Cucamonga (California), Dallas (Texas), Modesto (California), and Frankfort East (Kentucky). We will consider these five as our destinations.

We were only given eight months worth of sales because this is a relatively new project for Frito Lay and Gamesa. Based on this information we began our analysis.

3. Analysis of the Situation

Considering the different sources and destinations for each of the different products, we figured we needed to create a Multi-Commodity Transportation Model. Besides creating the algorithm in correct format on AMPL/CPLEX, we need some specific inputs to make the model work.

We need capacity bounds, if applicable, for the sources considered. In this case we assumed no bounds, therefore the manufacturing plants would have infinite capacity of production of these goods. Of course, in reality, we know that they do not actually have infinite capacity. Since these exports represent a minimal percentage of the plants' total outputs, and considering that these plants are able to increase production if necessary, the term infinite is convenient in our model for the purpose of our project.

We need a demand of each product for each destination point as well. Ideally, we would have created a forecast for future demands and use these for our model. But since we only have access to eight months of information, an accurate monthly forecast cannot be completed. We could not even look for seasonality trends due to the fact that we do not have a complete year of information. We also found out that the demand fluctuated a lot within each month. So based on all this, we used an average of the demands of each city for each product to get the demands needed for the model.

For the cost of each Arc, or the transportation cost between the cities, we got quotes from the truck company that moves products for Frito Lay, SWIFT.

4. Technical Description of the Model

For our Frito Lay project, we used AMPL/CPLEX to write a Multi Commodity Network Flow model and help us solve it. We had to write two files before running our model on AMPL. One of the files contained all of our data to be used and the other file was our model that would use our data file to solve the problem.

For our data file, we had a total of 7 nodes, which were the different points where our products would pass through or end up in. Our nodes were: Vallejo in Mexico, Dallas, TX, Obregon in Mexico, Cucamonga, CA, Modesto, CA, Pulaski, TN, and Frankfort, IN. Then we had to further specify what each node was. For doing that, we divided the nodes into two categories: sources and destinations. Vallejo, Dallas, and Obregon (Nodes 1, 2, and 3) were set as sources because our commodities would depart from these Nodes at some point in time. Dallas, Obregon, Cucamonga, Modesto, Pulaski, and Frankfort (Nodes 2, 3, 4, 5, 6, 7) were set as Destinations because they would all be receiving our products at some point in time.

We then had to define a set of Arcs in which the flow of our products would go through. There were a total of 10 arcs: from Vallejo to Dallas, Vallejo to Obregon, Dallas to Cucamonga, Dallas to Modesto, Dallas to Pulaski, Dallas to Frankfort, Obregon to Cucamonga, Obregon to Modesto, Obregon to Pulaski, and Obregon to Frankfort. For each arc, we had to determine a cost of transportation in this same data file. We got some of the costs directly from Frito Lay, and some other ones from a sales manager at Swift, Carmine Tirella. Some prices, for example, are $2,200.00 from Vallejo to Dallas, $1,838.59 from Vallejo to Obregon, and $4,511.00 from Dallas to Frankfort. (See FL_Data document annexed).

After that, we had to establish our commodities and their respective demands at each node. We have four products (commodities), which are: Grandma's Mini Fudge Chocolate Chips of 85g (commodity q), Grandma's Mini Fudge Chocolate Chips of 35g (commodity w), Munchies Mini Sandwich Crackers of 35g (commodity e), and Munchies Mini Sandwich Crackers of 85g (commodity r). So, in our data file, our four commodities or products are q, w, e, and r. For each product, we had to establish a parameter for their demands at each destination. We got the demands for each product from Frito Lay. They gave us an eight-month demand and we then did a demand forecast by taking the averages of the demands. We had to it with averages because there was not enough data to do an entire demand forecast. Our demands are given in dollars of gross revenue from each product. Once we got the demands for each product at each destination, we inputted that data into our data file. For example, the demands for the four products at Dallas were: $25,117.38 of product q, $6,957.48 of product w, $10,284.53 of product e, and $26,633.75 of product r. (See FL_Data document annexed).

Once we finished our data file, we had to write our code using our data file. We had to set Nodes, Sources, Destinations, and Commodities, which were all already defined in our data file. Then we had to set our Arcs and specify that the Arcs would only be from one Node to another. Then we had to set parameters for the Demand and the Cost. For demand, we specified that it would be at every node for every commodity. We also set the default value of the demands to zero in case that there is no demand for a given product at a given destination. For cost, we specified that each arc would carry its own cost. These costs were the ones previously established in the data file. We then loaded our data file into our code file so that we would be able to use that information.

Two variables were then established: x and bx. X is a variable that would tell us how much of each commodity would be sent in each arc. Bx, meaning binary x, is a variable representing two values, 0 or 1, that would let us know whether an arc, or route, would be used. If a value of 0 is given for an arc, it means that route would no be used, so there would be no flow of commodities on the arc. If a value of 1 is given, then it means that there is a flow of products on that arc.

Once the entire problem was set up, we had to write our objective function and its constraints. This function was written to minimize the cost of the total flow of products in the arcs. To solve the problem correctly, we had to write out three constraints that would allow all of the requirements to be fulfilled.

The first constraint deals with the sources. It says that each of the sources has to have an equivalent flow out than it has a flow in. All products are added to see the total production. All of these have to be less than or equal to production capability. Since, at current volumes of demand, the plants have the capability of supplying these demands, we just put a very big number as the productions capability, in this case being 999,999.

The second constraint deals with the destinations. It also says that flow out minus its flow in has to equal the demand of each product for each node. In other words, once that product has flowed in, then the amount of product that flows out has to be smaller by the amount of demand at that destination. This is a constraint to satisfy the demand at each node.

The third constraint deals with the arcs and the commodities. It says that the sum of all the cost of commodities in each one of the arcs has to be equal or less than a large capacity for each arc ($100,000,000) times a binary variable that represents arcs which arcs are needed. This will ultimately let us know which are the optimal arcs.

To solve the model, we ran our code file in AMPL and we got our results. Figure 1,2, and 3 show the code, data, and the result files, respectfully.

5. Analysis and Managerial Interpretation

The results of our model were somewhat unexpected to us. We thought that the products going to California would take the route through Obregon, and that the East Coast products would go through Dallas. Contrary to our predictions the model indicates that all the products should be consolidated in Dallas and shipped from there.

In Figure 3, we can see the results of the model. The first part says that the objective function value is 9,319.97. This is the total transportation cost per truck when satisfying all the demands. In our model, volume was not a constraint, so theoretically, all the products fit in one single truck (which is not possible). To make it more realistic, we had to compute the volume constraints by hand.

The first set of numbers and letters in Figure 3 (x[1,*,*] :) represent what commodities, along with their quantities, are transported between nodes 1-2 and 1-3. Node 1 is Vallejo plant, while nodes 2 and 3 are Dallas plant and Obregon plant, respectively. So, from commodity ‘e’, Munchies Mini Sandwich Crackers 35g , $60,742.30 worth of product should be transported from Vallejo to Dallas. For product ‘q’, Grandma's Mini Fudge Chocolate Chips 85g, $89,802.9, and so forth. The second set is considering the transportation from Dallas to the rest of the plants around the country. The third set represents the products passing through Obregon, which are clearly none.

Finally, the last set which starts with bx:= indicates the feasible arcs that will be used in the model. So, arcs 1-2, 2-4, 2-5, 2-6, and 2-7 will be used. The rest can be discarded. This means that Vallejo-Dallas, Dallas-Cucamonga, Dallas-Modesto, Dallas-Pulaski, and Dallas-Frankfort are the routs trucks will follow.

After checking the program was correct, we decided to consider different scenarios to further analyze Frito Lay's possible situations in the near future. We included unexpected and important changes in the demand for our products. For example, demand in California had an 80% increase. After running the model several times and with different (but realistic) demands, we found out results were the same; all products should be transported to Dallas first.

After further analysis we came across what we thought was the main difference between the two routes: the price on border-crossing fees and toll fees. Prior to crossing the Texas border through Laredo, trucks typically stop in Monterrey. On the other hand, in the Arizona border crossing through Nogales, trucks stop in Obregon. The total fees per truck in the Laredo border are $600, compared to the $1200 from the Nogales border. So, no matter how much demand plants in California have, it will always be cheaper to transport goods though Dallas. Since Dallas' plant has the capacity and personnel to consolidate, we recommend that all products are consolidated in this city and shipped to the rest of the destinations.

6. Conclusions and Critique

After having implemented our model to the problem, we concluded that all products should be consolidated in Dallas and from there be distributed to the other destinations. To do this, people in Vallejo are going to have to be trained in exporting (knowing how to fill out the proper documents, packaging, etc).

By implementing this new route, Frito Lay would be saving $73,926.20 per month and $887,114.40 per year as compared to their current routes. Frito Lay's total costs per month would now be $93,199.70 instead of their current cost of $167,125.90 per month.

A limitation to our model is that it does not have a constraint for the volumes that a truck has capacity for. Our model assumes that everything could fit in one truck. Once we got the results, we had to manually calculate how many truck would be needed to carry our volumes of product and multiply it by the total cost given by our model.

For further study, inputs in the model would have to change as demand changes. At our current demands, we had an "unlimited" supply. Once demands start growing, our supply constraint has to change to realistically include the production capacity of the plants. Also, if more destinations are going to be included in the distribution, then they also have to be inputted in the model with their respective costs. Transportation costs can change due to gas prices, renegotiation with the current transportation company, or a change in transportation companies used. Furthermore, when running in AMPL, the name of the file has to match both of the text files and, if modified, they have to be changed in the code and data files.

To conclude, we believe Frito Lay should change their currents transportation routes to the ones suggested by our model. That is, consolidate in Dallas.

7. Index

Figure 1

```
# Frito Lay Code
# Multi-Commodity Fixed Charge Network Flow Problem
reset;
set Nodes;
set Sources;
set Destinations;
set Commodities;
set Arcs within {Nodes, Nodes};

param Demand {Destinations, Commodities}, default 0.0;
param Cost {Arcs};

data e:FL_Data.txt;

var x {Arcs, Commodities} >= 0;
var bx {Arcs} binary;

minimize OBJ: sum{(i,j) in Arcs} Cost[i,j] * bx[i,j];

subject to C1{k in Sources, c in Commodities}:sum{(k,j) in Arcs} x[k,j,c] - sum{(i,k) in Arcs} x[i,k,c] <= 999999;

subject to C2{k in Destinations, c in Commodities}:sum{(k,j) in Arcs} x[k,j,c] - sum{(i,k) in Arcs} x[i,k,c] = Demand[k,c];

subject to C3{(i,j) in Arcs}: sum{c in Commodities} x[i,j,c] <= 100000000*bx[i,j];

solve;
display x, bx;
```

Figure 2

```
#Frito Lay Data File

# 1 is Vallejo,    # 2 is Dallas,    # 3 is Obregon,    # 4 is Cucamonga,    # 5 is Modesto
# 6 is Pulaski,    # 7 is Frankfort

# q Grandma's Minifudge Chocolate Chips 85g (99 cents)
# w Grandma's Minifudge Chocolate Chips 35g (33 cents)
# e Munchies Mini Sandwich Crackers 35g (33 cents)
# r Munchies Mini Sandwich Crackers 85g (99 cents)

set Nodes := 1 2 3 4 5 6 7;
set Sources := 1 2 3;
set Destinations := 2 3 4 5 6 7;
set Commodities := q w e r;
set Arcs := (1,2) (1,3) (2,4) (2,5) (2,6) (2,7) (3,4) (3,5) (3,6) (3,7);

param Cost :=

1 2  2200
1 3  1838.59
2 4  2074.66
2 5  2728.75
2 6  1053.60
2 7  1262.96
3 4  2461
3 5  3212
3 6  4730
3 7  4511;

param Demand :=

2 q -25117.38
2 w -6957.48
2 e -10284.53
2 r -26633.75
4 q -10343.83
4 w -5142.49
4 e -5434.41
4 r -10973.75
5 q -3685.02
5 w 0
5 e 0
5 r -5083.73
6 q -22828.19
6 w -23848.12
6 e -30438.61
6 r -24906.07
```

7 q -27828.52
7 w -12182.21
7 e -14584.75
7 r -32003.19;

Figure 3

```
CPLEX 10.0.0: optimal integer solution; objective 9319.97
0 MIP simplex iterations
0 branch-and-bound nodes

x [1,*,*]
:     e        q        r        w      :=
2   60742.3   89802.9   99600.5   48130.3
3      0         0         0         0

 [2,*,*]
:     e        q         r         w       :=
4   5434.41   10343.8   10973.8    5142.49
5      0      3685.02   5083.73      0
6   30438.6   22828.2   24906.1   23848.1
7   14584.8   27828.5   32003.2   12182.2

 [3,*,*]
:  e  q  r  w  :=
4  0  0  0  0
5  0  0  0  0
6  0  0  0  0
7  0  0  0  0;
bx :=
1 2   1
1 3   0
2 4   1
2 5   1
2 6   1
2 7   1
3 4   0
3 5   0
3 6   0
3 7   0
;
```