Strategic Delivery Route Scheduling for Small Geographic Areas

by

David Gilchrist

Submitted to the
Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of

Bachelor of Science in Mechanical Engineering as Recommended by the Department of
Mechanical Engineering

at the

Massachusetts Institute of Technology

June 2015

© 2015 Massachusetts Institute of Technology. All rights reserved.
Strategic Delivery Route Scheduling for Small Geographic Areas

by

David Gilchrist

Submitted to the Department of Mechanical Engineering on May 8, 2015 in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Mechanical Engineering as Recommended by the Department of Mechanical Engineering

ABSTRACT

The delivery scheduling process of a regional wholesaler was analyzed in order to develop a more strategic scheduling program. The strategic schedule was designed to utilize weekly demand history, opposed to daily demand, in order to decrease small batch deliveries, aid in store inventory management and foster customer relations. This was accomplished with a linear mathematical program, which produced a standard weekly schedule. A metric for the maximum days between deliveries was developed to show the improved delivery day distribution. For the 30 stores analyzed, the average maximum days between deliveries fell from 5.04 days to 3.37 days. The decreased time between deliveries will assist the small stores in inventory management. Additionally, the standardized schedule will allow storeowners and truck drivers to develop a productive relationship, which should be able to decrease delivery time and grow customer relations.

Thesis Supervisor: Dr. Chris Caplice
Title: Executive Director, Center for Transportation and Logistics
Acknowledgements

The author of this paper would like to thank Dr. Chris Caplice for his assistance in guiding him through this project, and Miguel Bethencourt for his corporate knowledge.
Table of Contents

Abstract 3
Acknowledgements 5
Table of Contents 7
1. Introduction 9
2. Background 11
   2.1 Delivery Routing, Schedule Optimization and Inventory Management 11
   2.2 Linear Programming 12
   2.3 Demand Variability and Pooling 12
   2.4 BeerCo 12
3. Formulation 14
4. Results and Discussion 16
5. Conclusion 19
Appendix: MATLAB Code 21
Bibliography 23
1. Introduction

Delivery routing can be a very complex problem that needs to be solved everyday by businesses across the globe. Typically, a linear program is used to minimize cost, which is subject to a number of constraints, such as truck capacity and store delivery windows. This paper studied a beverage wholesaler, which we will call BeerCo, who makes daily deliveries to the southeastern portion of Massachusetts. Each year, BeerCo’s trucks drive approximately 600,000 miles in order to deliver over 9 million cases of beer to 3,000 customers. The fuel costs alone amount to approximately $500,000, which ignores the cost of truck maintenance and manpower. Of the roughly 88,000 hours BeerCo drivers spend on the road, 63% of the time is spent servicing the stores, while the remaining 37% is spent driving between stops.

Prior to this study, BeerCo would take daily sales data and run it through commercial routing software in order to come up with the optimal delivery route and schedule for the next day. The software optimizes the routes by minimizing distance traveled and maximizing the quantity of goods on each delivery truck. The inherent problem with this system is twofold. At times it was found that stores were placing very small orders, on the magnitude of 5% of their weekly demand. This meant that in some weeks, stores were receiving multiple deliveries of small batches in addition to their bulk deliveries. The small deliveries are inherently inefficient and add to BeerCo’s overall logistics cost. Additionally, stores were receiving deliveries from multiple truck drivers. This prevented truck drivers and storeowners from developing beneficial business relationships. Many of these customers operate small stores with even smaller stockrooms. The job of the delivery driver is bring the product into the stockroom and rearrange it in such a way that older product is at the front and newer product is in the back. When a delivery driver gains the trust of the storeowner, and understands the particular difficulties in
delivering to each store, he is able to develop an efficient system to meet the needs of each store. This could result in decreased store service times for delivery drivers, which currently amounts to 56,000 man hours each year.

In order to address some of these concerns, a standardized delivery schedule was developed that could foster customer relations, decrease small batch orders and maintain if not increase overall delivery efficiency. The initial study identified geographically close customers that had relatively large weekly orders with low standard deviations. These customers offered the largest immediate benefit without added complexity. The ultimate goal was to create a system of strategic route scheduling that would incorporate inventory management and customer relations into the optimization.

The remainder of this thesis is organized as follows. Chapter Two outlines the background information needed to understand the thesis, which includes a basic explanation of linear programming, delivery routing and scheduling, inventory management, risk pooling as well as an introduction to BeerCo. In Chapter Three we describe the linear program used to solve the scheduling problem. Chapter Four describes the results of the linear optimization. In Chapter Five we provide conclusions of the research and areas of future research.
2. Background

2.1 Delivery Routing, Schedule Optimization and Inventory Management

Delivery size and store location are two of the most important factors in determining delivery routes and schedules. This study assumes that geographically clustered stores have been previously identified, which means that distance between stores can be ignored. On the other hand, the size of each delivery will have a serious impact on routing decisions. The most obvious impact of delivery size is on truck capacity. A truck can only deliver as much as its capacity dictates, which can prevent one truck from delivering to multiple large stores. Having a fleet of different sized trucks can help maximize truck utilization, but it is important to note that filling the largest trucks to capacity will result in the most efficient routes (Ballou 1992).

Reducing the number of deliveries to a store will always improve delivery cost, but it will not necessarily increase the effectiveness of the entire supply chain. Once a delivery has been made, the store is responsible for holding that inventory, which takes up valuable stockroom space. The typical BeerCo customer will carry hundreds if not thousands of different products, all of which are fighting for both shelving and stock room real estate. Due to store inventory constraints, it is sometimes not possible for a store to hold two or even one week of demand for all products. This constraint requires that many of the stores require a minimum of two deliveries each week, and some stores even more.

The days in which deliveries are made are also critical to proper inventory management. If a store receives two deliveries each week, but they are made on back-to-back days, the store will be holding six or more days of inventory. A store needs time to deplete inventory after a delivery is made to free up space for the next delivery. Thus, in the case of two delivery days per week it makes sense to spread the deliveries out over the week, the ideal case leaving a
maximum of three days between deliveries. In the case of three deliveries per week the optimal solution leaves at most two days between deliveries.

2.2 Linear Programing

Linear programming is tool used to determine the optimal solution to a problem described by linear relationships. An objective function drives the program to its solution by relating the decision variables in a way that a maximum or minimum solution will produce the optimal result. The decision variables are ultimately the outputs of the program, and are incrementally changed until an optimal solution is found. The decision variables can be related to a series of constraints in the form of mathematical equalities and inequalities. Additionally, decision variables can be integer, non-integer and binary.

2.3. Demand Variability and Pooling

An additional source of complexity in delivery routing is demand variability. Accurate forecasting models are hard to come by, which makes designing delivery routes more than a week in advance difficult. One method to combat demand variability is pooling, which occurs naturally on any delivery truck delivering to two or more stores. A delivery truck may deliver to a dozen stores each with an average weekly demand and standard deviation. Pooling is essentially a bet that a few of those 12 stores will be above average, while the others will be below average. This results in the “pooled” average having a lower standard deviation.

2.4 BeerCo

BeerCo services approximately 3,000 stores and restaurants in southeastern Massachusetts. Their product line is diverse and has high market share in the region. Their
customers range drastically in size from small restaurants that order a half dozen cases per week to large stores that process over 2,000 cases per week. BeerCo strives for a very high level of service, which has led them to use a flexible daily routing system. Software will process all sales orders overnight, and route them for next day delivery to the customer. This daily optimization makes for a highly flexible delivery network, but often ignores potential advantages that could be gained from weekly planning.
3. Formulation

Thirty of BeerCo’s customers were identified from the greater Boston area. These customers were chosen because of their shared geographic proximity and relatively consistent high demand. Four months of daily delivery data was examined and broken down by weekly demand, weekly deliveries and delivery size. Utilizing this data, a linear program was designed to create weekly-standardized schedules for the stores.

\[
\text{minimize: } \sum_i \sum_j Y_{ij}
\]

Subject to:

1. \( X_{ij} - M Y_{ij} \leq 0 \) for all \( i \) and \( j \)
2. \( \sum_i X_{ij} \leq C_j \) for all \( j \)
3. \( \sum_j X_{ij} = D_i \) for all \( i \)
4. \( \sum_j Y_{ij} \geq F_i \)
5. \( Y_{ij} + Y_{i(j+1)} \leq 1 \)

Where:

Indices:

- \( i \): The customer receiving the shipment, \( i = 1, 2, \ldots m \)
- \( j \): The day of the week, \( j = 1, 2, \ldots n \)

Decision Variables:

- \( X_{ij} \): The number of cases to be shipped to store \( i \) on day \( j \)
- \( Y_{ij} \): \( = 1 \) if store \( i \) will receive a shipment on day \( j \); \( = 0 \) otherwise

Data:

- \( M \): An arbitrarily large number relative to the maximum value of \( X_{ij} \)
- \( C_j \): The truck capacity for day \( j \)
- \( D_i \): The average weekly demand for store \( i \)
- \( F_i \): The minimum number of deliveries per week for store \( i \)

Figure 1: Mathematical formulation of the linear program
The objective function in Figure 1 that set out to minimizes the number of deliveries per week. In order to obtain this, two decision variables were required. The first was an integer variable, $Y_{ij}$, which took on the form of either 0 or 1. If $Y_{ij}$ was 1 it represented the fact that store $i$ received a delivery on day $j$, whereas a 0 represented no delivery. The other decision variable, $X_{ij}$, represented the volume of product being delivered in cases to store $i$ on day $j$. In order to represent the 30 stores and 5 delivery days, Monday through Friday, a total of 300 variables were required.

The first constraint is a forcing constraint used to link $Y_{ij}$ to $X_{ij}$. Or in other terms, Equation 1 was used to ensure that when $X_{ij}$ was nonzero, $Y_{ij}$ would equal one. Next, a daily truck capacity constraint, Equation 2, was used to limit daily shipments. Then, a weekly demand equality was used to ensure that each store would receive their average weekly demand, Equation 3.

Equation 4 assigned each store a minimum number of deliveries per week. While Equation 5 ensured that a store would not receive deliveries on back-to-back days. Both of these constraints were designed to help distribute inventory over the course of the week in order to assist stores in inventory management.

These 5 equations characterize the linear program used to create the standardized schedule. MATLAB software was then used to solve the program, which allowed for easy adjustments to the various constraints. The code used can be seen in the Appendix, MATLAB Code.
4. Results & Discussion

The linear programs easily produces a solution within the given constraints, but the benefits are not immediately clear. Each store receives two deliveries per week except for the three stores that have weekly demand of roughly 1,000 cases or more, who receive 3 deliveries per week. This results in a total 63 deliveries per week with approximately equal volume being shipped each day.

Table 1: An example of five stores weekly deliveries. All quantities are in units of cases.

<table>
<thead>
<tr>
<th>Delivery Day</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Store 1</td>
<td>341</td>
<td>0</td>
<td>342</td>
<td>0</td>
<td>341</td>
<td>1,024</td>
</tr>
<tr>
<td>Store 2</td>
<td>0</td>
<td>171</td>
<td>0</td>
<td>342</td>
<td>0</td>
<td>512</td>
</tr>
<tr>
<td>Store 3</td>
<td>168</td>
<td>0</td>
<td>0</td>
<td>336</td>
<td>0</td>
<td>504</td>
</tr>
<tr>
<td>Store 4</td>
<td>163</td>
<td>0</td>
<td>326</td>
<td>0</td>
<td>0</td>
<td>489</td>
</tr>
<tr>
<td>Store 5</td>
<td>0</td>
<td>234</td>
<td>0</td>
<td>0</td>
<td>116</td>
<td>350</td>
</tr>
<tr>
<td>Daily Total</td>
<td>672</td>
<td>405</td>
<td>668</td>
<td>678</td>
<td>458</td>
<td>5,763</td>
</tr>
</tbody>
</table>

As seen in Table 1, the delivery days for each store are spread throughout the week. The metric used to quantify this distribution is the maximum number of days between deliveries in the week. For instance, if a store received deliveries on Monday and Thursday the maximum gap between deliveries would be Friday to Sunday, which is three days. A lower maximum represents a better-distributed schedule, which results in improved inventory management at the store. The program’s solution produced an average maximum of 3.37 days between deliveries for the 30 stores. In order to compare this metric to the previous system, four random weeks were selected from the historical data, one from each month of data to eliminate potential seasonal trends. The average maximum time between deliveries over these four weeks was 5.04 days.
Table 2: The average maximum number of days between deliveries for historical data and the newly generated schedule

<table>
<thead>
<tr>
<th>New Schedule</th>
<th>September</th>
<th>October</th>
<th>November</th>
<th>December</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.37 Days</td>
<td>5.17 Days</td>
<td>5.07 Days</td>
<td>5.40 Days</td>
<td>4.43 Days</td>
<td>5.04 Days</td>
</tr>
</tbody>
</table>

Figure 2, below, shows the same data in histogram form. It can be seen that the historical data, in red, is comprised mostly one-day delivery weeks, which leaves six days between deliveries. On the right side, the optimized solution shows that the majority of stores receive multiple deliveries spread out over the course of the week.

![Histogram](image)

Figure 2: Histogram of the maximum days between deliveries in the historical data in red. The red bars represent historical data, while the blue bars represent the proposed solution.

Thus far, the optimization has assumed that each store receives the average weekly demand each week, but this is not a realistic scenario. The first way that the model accounts for this variability is through pooling. Each delivery day in the standardized route schedule has an average of 12.60 stops, which means that on average 12 stores are being delivered to each day.
Operating under the assumption that the variance in demand is uncorrelated, the average demand for each delivery day should remain fairly constant. The stores with below average demand will leave capacity for the stores with higher than average demand. As the number of stops per day increases, the effects of pooling should decrease daily demand variability.

**Table 3:** The average deliveries per day for the proposed route compared to the average deliveries per day of the historical data

<table>
<thead>
<tr>
<th></th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Historical</td>
<td>5.00</td>
<td>7.00</td>
<td>11.50</td>
<td>5.00</td>
<td>12.00</td>
<td>8.10</td>
</tr>
<tr>
<td>Proposed</td>
<td>13.00</td>
<td>15.00</td>
<td>9.00</td>
<td>10.00</td>
<td>16.00</td>
<td>12.60</td>
</tr>
</tbody>
</table>

The second way that demand variability was addressed was through sensitivity analysis. The model was run under the extreme case that one half of the stores received their average demand and the other half received their average demand plus one standard deviation. This represents the most extreme case seen in historical data were the weekly demand for the 30 stores approached 21,000 cases, opposed to the average 16,000 cases. This situation would require added truck capacity, but the model was able to distribute this need evenly over the week. This result will allow BeerCo to assign larger trucks to certain drivers over the course of the week, instead of adding a new route. Lastly, it is important to note that the proposed delivery volumes represent the average delivery size, and should be used as the middle point in a range of values, opposed to exact quantity.
5. Conclusion

This study set out to create a method for strategic delivery route scheduling. It was found that through the use of linear programming, a weekly delivery route schedule could be developed that decreases the average days between deliveries from 5.04 days to 3.37 days. This allows storeowners to better manage their stock room inventory, by lowering their average daily inventory. Additionally, the proposed schedule has the potential to decrease service times by increasing delivery driver customer knowledge. The standardized schedule results in drivers servicing the same stores each week, which allows them to customize their delivery process at each store. Lastly, the standardized schedule will allow storeowners and sales representatives to plan deliveries over the course of the week, which should eliminate the need for last minute small batch orders.

It is important to note that this strategy would not be optimal if applied to 100% of the delivery volume. By implementing this approach to the total average demand minus one standard deviation, a wholesaler could maintain a high truck capacity utilization, and therefore maintain delivery route efficiency. If this strategy were implemented to all demand, capacity utilization would drop, because the schedule would not be able to account for variation in demand.

If future investigation into this topic were conducted, there are several areas that would warrant exploration. First, the schedule was just recently implemented on a small scale at BeerCo, which means the impact on delivery driver service time has not been investigated. Once these data becomes available, it would be beneficial to determine whether or not service time decreases. Additionally, this study focused on only 1% of BeerCo’s customers. There is still much to be understood on the impact of overall logistics costs if this were to be applied to a
majority of the customers. For instance, the percentage of volume that could be handled using this strategy without impacting truck capacity utilization.
Appendix: MATLAB Code

% Objective functions components
y_ij = repmat(1,150,1);
x_ij = zeros(150,1);

%objective function
f = [y_ij; x_ij];

%lower bound of x and y
lb = zeros(300,1);

%upper bound components
uby = repmat(1,150,1);
ubx = 1.4*maxdelivery; %upper bound of each delivery relative to weekly demand

% upper bound of all variables
ub = [uby; ubx];

% range of integer variables
intcon = [1:150];

% formulation of inequality coefficient matrix

%daily capacity constraint
one_30 = repmat(1,1,30);
mon_demand = [zeros(1,150), one_30, zeros(1,120)];
tues_demand = [zeros(1,180), one_30, zeros(1,90)];
wed_demand = [zeros(1,210), one_30, zeros(1,60)];
thurs_demand = [zeros(1,240), one_30, zeros(1,30)];
fri_demand = [zeros(1,270), one_30];
A_daily_capacity = [mon_demand; tues_demand; wed_demand; thurs_demand; fri_demand];

%min delivery day constraint
A_delivery_day = -mindeliveryconstraint';

%correlate x and y
A_x = diag(repmat(-2000,1,150));
A_y = diag(repmat(1,1,150));
A_xy = vertcat(A_x,A_y);

%non consecutive delivery days
A_nonconsec = StoresandAveragesS2';

%inequality coefficient matrix, A
A = vertcat(A_daily_capacity,A_delivery_day,A_xy', A_nonconsec);

%inequality vector, B
B = [repmat(4400,1,5), repmat(-2,1,30), repmat(0,1,150),repmat(1,1,120)]';

%equality coefficient matrix, AEQ
AEQ = weeklydemandconstraint';
%equality vector, BEQ
BEQ = halfstd(1:30);

%solution
[solution,fval,exitflag,output] = intlinprog(f,intcon,A,B,AEQ,BEQ,lb,ub);
References