A Stochastic Approach to the Burrito Problem*

Lillian Makhoul Abigail Nix

May 2024

Abstract

In this project, we explored the problem presented in Gurobi's Burrito Optimization Game. The problem provides a set of customer locations and possible truck locations, as well as values for the demand for burritos at each customer location. By strategically placing trucks, the goal is to maximize the profit of the burrito trucks (revenue from burritos sold minus cost of placing and operating the trucks). Gurobi solves this problem as a deterministic linear program, which will be introduced in the next section. For this project, we chose to expand on this deterministic model of the burrito problem and include uncertainty in the demand parameters. To do this, we first introduce stochastic programming and the two-stage model more generally, and then show how we used this theoretical model to create a model for the burrito problem with uncertain demand. Finally, we will present the results of our model and compare them with the results Gurobi gets in the actual Burrito Optimization Game, as well as using Gurobi's LP model.

1 Introduction

The burrito problem [3] is an example of an uncapacitated facility location problem (UFLP), where uncapacitated here refers to the fact that each burrito truck can serve unlimited customers. In this section, we will introduce the deterministic problem in order to later adjust this model to account for uncertainty in the problem set up.

1.1 The Original Model Formulation of the Burrito Problem

Before adding uncertainty to the problem, we present the deterministic model for the game, as provided by Gurobi. The following variables and parameters are used in this model:

Sets

C: the set of customer locations

L: the set of potential locations for placing a food truck

Parameters

r: the revenue (in \$) from selling one burrito

k: the cost in (in \$) of ingredients for one burrito

f_j: the fixed cost (in \$) of placing a truck at location j, $j \in L$

d_i: the demand for burritos at location i, $i \in C$

$\alpha_{i,j}$: the fraction of demand that will be captured from customer location i by placing a truck at location j, $i \in C$, $j \in L$

Decision Variables

x_j: $\begin{cases} 1 & \text{if we place a truck at location } j, \\ 0 & \text{otherwise} \end{cases}$, $j \in L$

*The numerical results generated in this report are associated with the code in https://github.com/makhoullillian153/A-Stochastic-Approach-on-the-Burrito-Problem

$$y_{i,j}: \begin{cases} 1 & \text{if the demand at customer location } i \text{ is assigned to location } j, \\ 0 & \text{otherwise} \end{cases} \quad , i \in C,\ j \in L$$

The Model

With these all defined, we can formulate the model itself.

$$\begin{aligned} \text{Maximize} \quad & \sum_{i \in C} \sum_{j \in L} (r-k)\alpha_{i,j} d_i y_{i,j} - \sum_{j \in L} f_j x_j & & (1) \\ \text{s.t.} \quad & \sum_{j \in L} y_{i,j} \leq 1 \qquad i \in C & & (2) \\ & y_{i,j} \leq x_j \qquad i \in C,\ j \in L & & (3) \\ & x_j, y_{i,j} \in \{0,1\} \qquad i \in C,\ j \in L & & (4) \end{aligned}$$

Here, the objective function that we want to maximize is the sum of all the profit gained from selling burritos (revenue minus ingredient cost for each burrito sold) minus the fixed cost of placing burrito trucks. Our first constraint, (2), declares that customers from location i can go to at most one burrito truck location j, i.e., customers go to the closest burrito truck. The second constraint, (3), ensures that customers only go to burrito truck locations that are open, i.e., customers cannot buy a burrito from a location with no burrito truck. The final constraint, (4), simply says that x_j and $y_{i,j}$ are decision variables.

2 Bringing Uncertainty to the UFLP

In Gurobi's Burrito Optimization Game, the first round is the problem set up we described in the previous section. However, the second round of the game introduces uncertainty in the demand for burritos at each customer location. In a real world setting, the demand at each customer location would not be known exactly, and instead we would only have a forecast for what the demand would look like and some ranges for error. This forecast could be determined from past data for demand at that location, and could also include other factors. Because there is no way to know with certainty the actual demand at a location, the model becomes much more realistic if we make demand uncertain. The next model that we will demonstrate does exactly this, using a two-stage model to represent the new stochastic problem.

2.1 Stochastic Programming and the Two-Stage Model

In order to incorporate some uncertainty into the demand of our problem, we will use some key ideas in stochastic programming. In general, a stochastic program is a type of optimization problem where some number of parameters are uncertain, but, we assume that the probability distributions of these parameters are known. Although in a real world setting, these distributions may not actually be known, incorporating this kind of uncertainty still makes the problem more realistic while maintaining the ability to formulate and solve a model for the problem.

One of the most common types of stochastic programming models is called the two-stage model, and this is the type of model we use for the stochastic burrito optimization problem. The general two-stage model is as follows [1]:

$$\begin{aligned} \text{Minimize} \quad & f(x) + \mathbb{E}_\xi Q(x, \xi(\omega)) \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0, \end{aligned}$$

where ω is an outcome such that $\xi(\omega)$ is a particular realization of the random vector ξ, and $Q(x, \xi(\omega))$ represents the optimal value of the second stage problem:

$$\begin{aligned} \text{Minimize} \quad & q(y, \xi(\omega)) \\ \text{s.t.} \quad & Wy + T(\omega)x = h(\omega) \\ & y \geq 0. \end{aligned}$$

Here, $q(y, \xi(\omega))$ represents the second stage problem, and $Q(x, \xi(\omega))$ is the optimal value of that problem. In the generic two-stage model, there are both first stage decisions and second stage decisions, which correspond to the two parts of the model. Here, x represents the first stage variable (or first-stage decision), and y represents the second stage variable (or second-stage decision, i.e, correcting decision).

2.2 Applying Sampling Average Approximation

The first stage is before the random vector, ξ, is known, and then the second stage uses different possible realizations of ξ to optimize the second stage problem. To use ξ in the second stage problem, we use a Monte Carlo sampling method. We create a fixed number of different scenarios (different realizations of ξ) and name them $\xi_1, \ldots, \xi_V$. Each of these scenarios corresponds to its own subproblem. Then, to approximate the expected value of $Q(x, \xi)$, we use the sample average approximation method, i.e., we calculate

$$\frac{1}{V} \sum_{k=1}^{V} Q(x, \xi_k).$$

Here, we assume that each of the generated realizations of ξ, i.e., the different scenarios, are equally likely. This value is then used in place of the original expected value in the model, and we solve the new first stage problem

$$\begin{aligned} \text{Minimize } & c^T x + \frac{1}{V} \sum_{k=1}^{V} q_k^T y_k \\ \text{s.t. } & Ax = b \\ & W y_k + T_k x = d_k \\ & x,\ y_k \geq 0. \end{aligned}$$

Now, we have formulated a deterministic problem where as our value for V increases, our optimal value of this deterministic program will approach that of a two-stage program [1]. Note that the constraints for this deterministic problem combine the constraints for the first stage and second stage problems in the original model.

Note: While this project focuses exclusively on the two-stage model as a method of integrating uncertainty into our problem, there are other ways to do this. For example, we could also use robust optimization, probabilistic programming, or formulating a deterministic problem using expected value of the random parameters. More information on these methods can be found in [1] and [2]. Each of these methods could result in different optimal solutions and may make more or less sense depending on the problem.

3 Model Formulation

To establish our two-stage model, we first need to determine which parameters and variables correspond to the first stage problem, and which correspond to the second stage problem. Since we are making demand uncertain, this parameter will need to be included in the second stage, along with the other parameters and variables that appear with d_i in the original model. This gives us the following:

First-stage Parameters and Variables

f_j: the fixed cost of placing a truck at location $j \in L$

x_j: $\begin{cases} 1 & \text{if we place a truck at location } j, \\ 0 & \text{otherwise.} \end{cases}$, $j \in L$

Second-stage Parameters and Variables

B: Number of scenarios

r: the revenue (in \$) from selling one burrito

k: the cost in (in \$) of ingredients for one burrito

d_i^l, d_i^h: lower and upper bounds for generating a random demand at location $i \in C$.

$\xi_{i,b}$: Random variable of demand for location $i \in C$ for each scenario b, $b = 1, \ldots, B$, where $\xi_{i,b} \sim \mathcal{U}(d_i^l, d_i^h)$.

$$y_{i,j,b} = \begin{cases} 1 & \text{if the demand at customer location } i \text{ is assigned} \\ & \text{to location } j \text{ for scenario } b, \\ 0 & \text{otherwise.} \end{cases} \qquad , i \in C,\ j \in L, b = 1, \ldots, B$$

The Model

The model itself follows the template of the classical two-stage model. For our second stage problem, we use Monte Carlo sampling to get the objective function. This forms a deterministic model rather than two separate first stage and second stage models. The model is provided below:

$$\begin{aligned} \text{Maximize} \quad & \frac{1}{B} \sum_{b=1}^{B} \sum_{i \in C} \sum_{j \in L} (r-k)\alpha_{i,j} d(\xi_{i,b}) y_{i,j,b} - \sum_{j=1}^{n} f_j x_j \\ \text{s.t.} \quad & \sum_{j \in L} y_{i,j,b} \leq 1 && i \in C, \quad b = 1, \ldots, B \\ & y_{i,j,b} \leq x_j && i \in C, \quad j \in L, \quad b = 1, \ldots, B \\ & x_j \in \{0,1\} && j \in L \\ & y_{i,j,b} \in \{0,1\} && i \in C, \quad j \in L, \quad b = 1, \ldots, B \end{aligned}$$

4 Results

Suppose the number of customer locations is three, and the number of burrito truck locations is eight. Using the model provided by Gurobi, we found the optimal profit to be approximately \$85. When setting the seed to the same number (as to generate the same values for demand and α), we found our output to be approximately \$103.

What explains this discrepancy? Looking closer at the generated data, we see the first two locations have a demand of 41.7 and 72, with the range for generating ξ of (41.3,42.6) and (70.3,76.2). However, the demand for the third location is very close to zero, yet has a range for generating ξ of (0,9.6). In the original formulation, a burrito truck would practically make no profit when placed near this third location. Now, the demand may increase by up to 9.6 points, and now we may make a proper profit off of this location. This in turn leads the model to finding an optimal profit which is noticeably higher than that of the original formulation.

Suppose we would like to look at a case where this does not occur. We keep the same bounds for our demands, but we fix the third location's average demand to be 50.

Now, Gurobi's model generates an optimal profit of approximately \$327, and the stochastic model gives an optimal profit of about \$333.

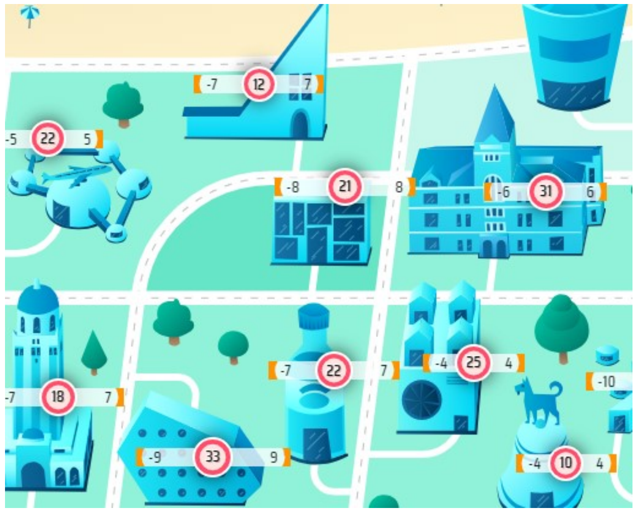
It is also important to note that three is a very small number. As we increase the number of customer locations, the optimal profit will not be as sensitive to a dramatic change in demand.

Suppose we now have 25 customer locations and 70 burrito locations. Gurobi's model yields an optimal profit of \$4232 , and our model yields an optimal profit of \$4168. As the problem grows, the discrepancy between Gurobi's optimal profit and our model's solution becomes more negligible! In this case it was about 1.2% of the profit of the original problem, whereas before it was about 21%.

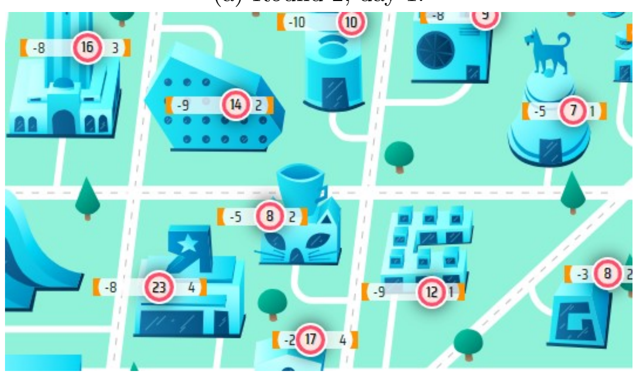
4.1 Applying Our Model to the Game

What would be more interesting is to see how our model competes in round of the Burrito Game! For this, we opened up round 2 day 1 (Figure 1a), where we now have uncertainty in the demand forecast. We hard coded all of the data into our program and placed trucks according to the found solution. And... "Your solution is as good as Gurobi's"!

We also looked into day 4 (Figure 1b), where our distribution isn't so symmetrical. Once again, we obtained the same result as Gurobi's.



(a) Round 2, day 1.



(b) Round 2, day 4.

References

[1] John R. Birge and François Louveaux. *Introduction to Stochastic Programming*. Springer New York, NY, 2011. ISBN: 978-1-4939-3703-5.

[2] Chun Cheng, Yossiri Adulyasak, and Louis-Martin Rousseau. "Robust facility location under demand uncertainty and facility disruptions". In: *Omega* 103 (2021), p. 102429. ISSN: 0305-0483.

[3] Gurobi Optimization. *Burrito Optimization Game: Game Guide*. URL: `https://www.gurobi.com/burrito-optimization-game-guide/`.