GUROBI OPTIMIZATION

BURRITO OPTIMIZATION GAME: Sample Lesson Plan

GUROBI: ALWAYS FREE FOR ACADEMICS

BURRITO© OPTIMIZATION GAME

www.burritooptimizationgame.com

A FREE EDUCATIONAL TOOL THAT INTRODUCES PLAYERS TO THE POWER OF OPTIMIZATION.

Overview

The Burrito Optimization Game is an educational game designed to introduce students to the power of optimization. In the game, the player places burrito trucks on a city map in order to earn as much profit as they can. In playing the game, the player is essentially solving an optimization problem "by hand." The game is designed to introduce players to the concept of optimization—what it is, what it's useful for, and why it's hard to do by hand.

This lesson plan is designed for a 45- to 60-minute class. Feel free to adjust the plan for shorter or longer classes.

The Burrito Optimization Game was produced by Gurobi in collaboration with Professor Larry Snyder of Lehigh University.

Audience

This lesson is designed for students who are knowledgeable about data science or computer science, but who haven't had much exposure to mathematical optimization.

Learning Objectives

If nothing else, players should come away from the game having learned these three things:

- **Planning doesn't end with forecasting**. Even if you have perfect forecasts, making optimal decisions with those forecasts is challenging.
- **Optimization by hand is hard**. Even for the small "city" in the game, it's hard to optimize by trial and error. For more realistic datasets, optimizing by hand would be impossible.
- **Optimization can be done by algorithms**. Mathematical optimization is a mature scientific field with robust commercial and open-source software.

Requirements to Play

Each student will need a computer with Internet access during the lesson. Desktop or laptop computers are fine, but the game does not work on tablets or phones. The recommended browser is Chrome. Students may share computers if desired, but we recommend no more than two students per computer.

Before Class

Prior to the lesson, students should:

- Register for a free Gurobi account at www.BurritoOptimizationGame.com. (Scroll down to "Registration".) Registration takes about 90 seconds.
- After registering, students can choose to click "Play Game" and begin to get a feel for how the game works.

Prior to the lesson, the instructor should:

- Set up a Championship, if desired. For step-by-step instructions, see the online Teaching Guide.

In Class

1. **Introduction.** Introduce the game itself, including the object of the game, rules, and game play mechanism. You can use the template slides available at www.burritooptimizationgame.com as a starting point and/or demonstrate the game live as you play it and introduce the main concepts.

2. **Play 2-3 "days".** Have the students play 2 or 3 "days" to get a feel for the game.

3. **Discussion.** Pause the game play to discuss with the students. What was easy? What was hard? Did anyone get close to optimal? Any questions about the mechanics of the game? Did the tips (light bulbs) help you improve your strategy?

4. **Finish Round 1.** Have the students finish Round 1 (5 days) and then stop before moving on to Round 2. (Alternately, you may ask students to click the "Restart" button to go back to Day 1, treating the initial play as a dry run.)

5. **Discussion.** How did you do? Was it easy/hard/fun/frustrating/confusing? How did you decide where to locate burrito trucks? If you were going to design an algorithm to automate the process of locating burrito trucks, how might your algorithm work? What are all those nerdy names for the buildings all about? *In addition, use this discussion to highlight a few of the "aha!" moments discussed in the Appendix.*

6. **Uncertainty or optimization.** At this point, the lesson can go in one of two directions:

   a. **Coping with Uncertainty**. Have students play Round 2 (5 more days). Round 2 introduces uncertainty into the game, where players no longer know the exact customer demands before they happen. Instead, players are given demand forecasts, including "error bars", that help players understand the range and skew of the uncertain demands. After they have finished, discuss the ways that uncertainty affected their solutions, and discuss possible ways to approach uncertainty when optimizing. (See "Coping with Uncertainty" section in the slides.)

   b. **Optimization 101.** Provide a quick overview of optimization: Explain why brute-force approaches (enumeration) will not work. Show the mixed-integer programming (MIP) formulation of the burrito-truck location problem, discussing it in whatever level of depth you wish. Explain the basic concept of branch-and-bound, in whatever level of depth you wish. Mention the availability of software for mathematical optimization. (See the "Optimization 101" section in the slides.)

7. **Discussion.** Wrapping up: What are some projects you have worked on that might benefit from optimization? What real-life applications do you think might use optimization (some examples we like: meal delivery planning, Google maps, etc.)? What topics would you like to learn more about?

After Class

Championship Mode

For homework, you may have students compete against each other (and you!) in Championship Mode. In Championship Mode, players compete to find the best solutions, and the players with the best total scores are listed on a Leaderboard. Championship Mode uses different datasets than the normal play mode, and the optimal solutions are kept secret. For instructions on how to set up a championship, see the online Teaching Guide.

Additional Reading

If you'd like to give the students something to read to provide more details about the optimization aspects of the game, we suggest the Optimal Solutions section of the online Game Guide. Additional references are listed in the Appendix.

Burritos!

You and your students will definitely want burritos for dinner. We can't help with that part. But enjoy!

Appendix: "Aha!" Moments

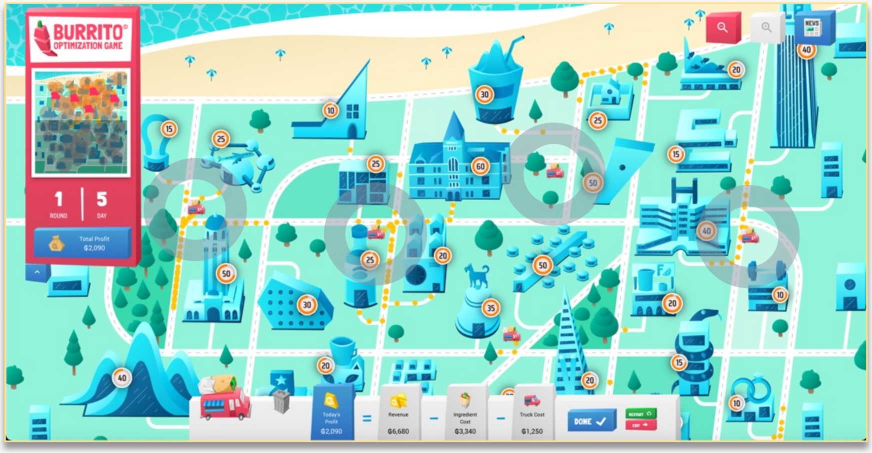
There are many insights that you can highlight when your students are playing the game. Some of these will occur organically and you can discuss them when they do; others will require you to provide a bit more instruction to lead students to those moments. Some examples are discussed below.

Note: You can skip directly to a certain round or day by clicking the gear icon in the top-right corner of the game landing page (instead of clicking "Play the Game"). This is useful when demonstrating some of the learning lessons.

Tradeoffs

In the game, there is a tradeoff between (fixed) truck costs and (variable) revenues: The more trucks you open, the more you pay in truck costs but also the more net revenue you earn (and the more ingredient cost you pay).There is more than one way to navigate this tradeoff. However, it might be worth taking a look at the minimum number of customers a truck needs to feed to break even (recoup the cost of the truck placed). For example, the first solution shown below uses two trucks, the second uses four, but they have nearly the same total profit.

Ex: [Truck Cost (₲/truck)] / [Profit per customer (₲/customer)] = 250 ₲/truck / (10 - 5 ₲/customer) = 50 Customers/truck to break even)].



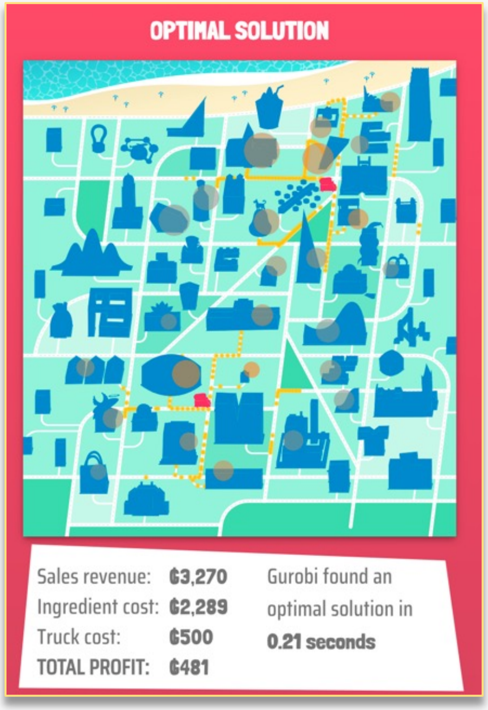
Diminishing Returns

The first truck you place in a given part of the city will usually have a big impact on your profit. The second truck will have a smaller impact, and subsequent trucks will each tend to have less impact, because the area becomes "saturated"—there is not enough demand to support the additional trucks. In the solutions pictured, the customers near the bottom of the first solution are already fairly well served; the second truck (in the second solution) will capture those customers but might not bring enough extra revenue.

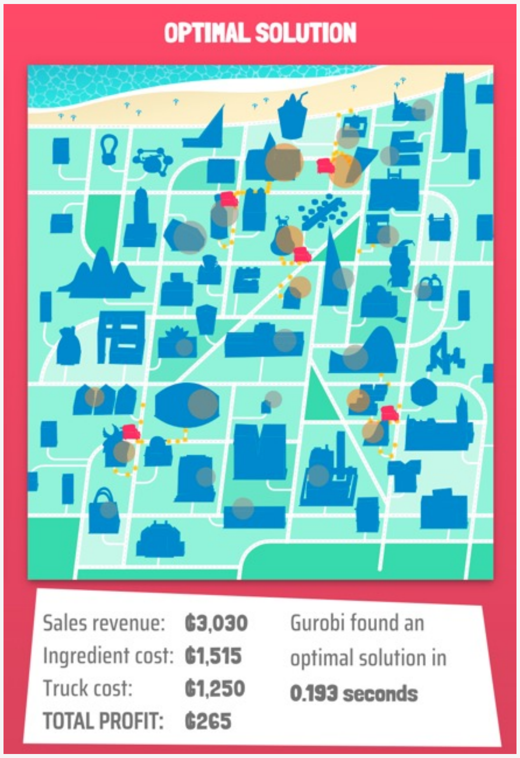
Optimal Solution Depends on Parameters

Although optimization is sometimes viewed as a black box by people new to the topic, optimal solutions still behave in predictable ways that students can anticipate. For example, if the ingredient cost per burrito increases, do you expect that the optimal profit will increase or decrease? (It shouldn't be too hard for the class to realize that the optimal profit should decrease.) What about the optimal number of trucks—will that increase or decrease? (If the profit per burrito decreases, it takes more customers to offset the fixed cost of each truck.)

Therefore, the optimal number of trucks will probably decrease.This is more subtle and will probably require more guidance from the instructor for students to see it.) The optimal solution under the default settings uses 4 trucks (left-hand image), but when the ingredient costs double, the optimal solution uses 2 trucks (right-hand image).

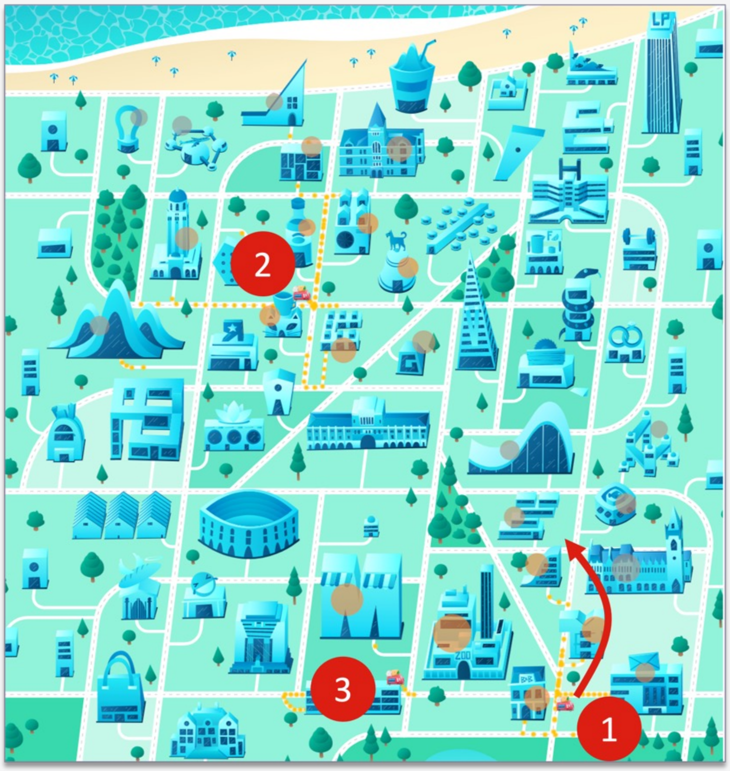


On Day 3, the newsfeed says that it's rainy, so customers are not willing to walk as far to buy burritos. Do we expect the optimal number of trucks to increase or decrease? This one is trickier: One could argue that, since each truck captures fewer customers, each truck is less profitable, and therefore we should open fewer trucks. Or one could argue that we should open more trucks since each one serves a smaller radius of buildings. The answer actually depends on the specific data: If the demand at each building is sufficiently large, it justifies opening more trucks to capture that demand, while if the demands are small, then it will be optimal to open fewer, or even zero, trucks. But, in this case the demands are large enough that it is optimal to open more trucks. The optimal solution has 5 trucks (right-hand image), whereas the optimal solution under the default setting has 4 (left-hand image).



Greedy May Not Be Optimal

Many players take a greedy approach to locating trucks in the Burrito Optimization Game: First, open the truck at the location that gives the largest profit; then keep that truck open, find the location that increases the profit the most, and open a truck there; and continue until adding a truck at any spot would decrease the profit. The greedy approach is not optimal for this problem (it is a heuristic—an approximate algorithm), although it is optimal for some problems, like the minimum spanning tree problem. For example, in the image below, the greedy approach would locate a truck at the spot marked 1, then 2, then 3, but then the original truck at spot 1 is no longer optimal; it is optimal to move it north a few blocks, as indicated on the figure. In other words, we cannot find the optimal solution using the greedy approach in this case.



How Might an Algorithm Work?

If you were going to design an algorithm to solve the burrito-truck problem, how would your algorithm work? Students should brainstorm possible approaches. One way students can come up with an approach is to think about the steps that they took when they were trying to solve the problem, and then try to generalize that approach so that it can be automated. Possible responses from students might be:

- A greedy approach, as discussed above
- Some sort of partitioning approach: solve the problem in each quadrant of the city separately and then combine the solutions
- Enumeration: program the computer to try every possible combination of truck locations

If enumeration enters the discussion, it presents a good opportunity to discuss why this approach is doomed to failure. For example, suppose we had a computer that could evaluate 1 billion solutions per second. Our computer could enumerate the solutions in Round 1, Day 1 very quickly: The problem has 16 allowable truck locations, so 2^{16}=65,536 solutions to evaluate, which our computer could do in less than 1 millisecond. But on Day 5 there are 56 locations and 2^{56}=72,057,594,037,927,900 solutions, which would take our computer 2.3 years! And if the problem had 100 locations, enumeration would take more than 40 trillion years! This motivates why we need smarter optimization algorithms, such as those that are "under the hood" of Gurobi or other exact solvers.

Uncertainty Makes Optimization Harder

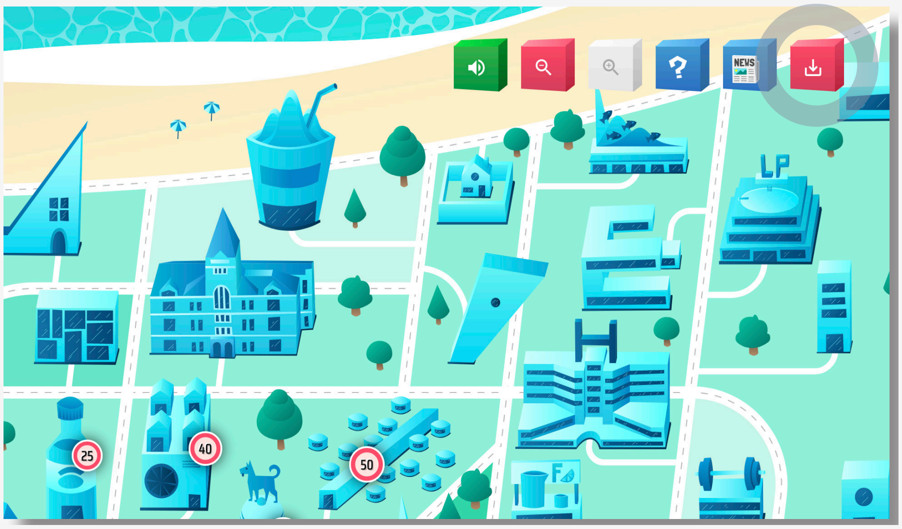
In Round 2, the demands become random: The player only knows forecasts of the demands when they choose truck locations (as does Gurobi), but the player's solutions (and Gurobi's) are evaluated using actual demands. Solutions that seemed good under the forecasts might perform quite poorly under actual demands. For example, the truck in the southwest quadrant of the city in the solution below looked attractive under the demand forecasts (dashed circles), which were large, but was probably not profitable under the actual demands (filled circles). It is even possible for the player's solution to be better than Gurobi's optimal solution in Round 2, because the solutions are optimized under one objective (expected profit) but evaluated under another (actual profit).

Forecast Bias Changes the Solution

On Days 4 and 5 of Round 2, error bars indicate the minimum and maximum possible values of the demands, and these are not always symmetrically spread around the forecast. Discuss whether we can make use of that information when planning truck locations. For example, if given a choice, should we locate in regions where the actual demands are likely to be higher than the forecasts, or lower? Students should realize that we would rather locate in regions where the actual demands are likely to be higher than the forecasts—where there is more upside potential and less downside risk.

Appendix: Solving the Problem Using Gurobi

Coming soon.



Appendix: Additional Resources

For students looking to learn more, here are some resources to help:

- To get your no cost Gurobi Academic License, apply on the Gurobi website and watch this video tutorial.
- To get started with Gurobi and optimization basics, see Gurobi's Resources for Beginners.
- For more optimization books and recommended blogs, see additional optimization-related resources.