

Mini-Solution Assignment

A large retailer, Wally World (WW), has 10 distribution centers and 1100 retail stores and it wants to use optimization to minimize its annual cost of transportation. WW has a policy of serving each of its stores from one and only one distribution center (DC), and it must also pay attention not to exceed the trailer capacity of its DCs, which is measured in the number of trailers per year that can be loaded. WW managers must also ensure that each retail store receives the number of trailers per week that are required to keep shelves stocked.

Data regarding the DCs and their capacity is contained in the `mini_solution` database, in particular, in the `dcs` table. Similarly, the `stores` table in that same database specifies how many trailers per year are required by each store. The database also has a `mileage` table for the distance between each pairing of DCs and Stores. The cost of each trip is \$200 fixed cost per trip plus \$0.75 per mile.

As a preliminary step you will need to import the `mini_solution` database into MySQL Workbench by using the command `File>Run>Script` on the `mini_solution.sql` file.

Problem 1

- 1) Write a formulation for this problem using Greek notation, using summation signs being careful to define all your decision variables, data, and parameters.
- 2) Write a Python program that optimizes this problem using Gurobi. Retrieve the data from the database mentioned above using a stored procedure in order to construct the optimization problem in Python-Gurobi.
- 3) Write a stored procedure to input the results back into the `mini_solution` database to record the number of trailers loads that go from each DC to each Store.
- 4) In the console window print the following data:
 - a) The optimal objective function value
 - b) Print a list of tuples (`dc_id`, `store_id`) where `dc_id` is the index of the DC that serves a Store with an index of `store_id`. Only print out those tuples where a particular DC serves a particular store.
 - c) Implement a timer to determine how long Gurobi took to solve the problem, that is, start the timer before the `m.optimize()` command and stop it after that statement is executed. Print out the elapsed time in seconds.

Problem 2

One of the data scientists at WW wanted to find a way to reduce the amount of time required to solve this optimization problem. She hypothesized that if the optimization allowed only the three or four closest DCs to serve each store then the optimization would run faster because there would be many fewer decision variables. Investigate this hypothesis by doing the following:

- 1) Write a stored procedure to determine the DCs closest to each store and create decision variables only for those pairings. Implement a timer for the Gurobi optimization in your program as you did in Problem 1. Run this program twice, once when the closes 3 DCs are

considered for each store and once when the closest 4 DCs are allowed to serve each store. (This requirement to solve multiple problems should make you think that your stored procedure to retrieve problem data should have a parameter in it for choosing 3, 4, or however many closest DCs are desired. Also, think of using the LIMIT clause in a stored procedure.)

- 2) Compare the time required for solving Problem 2 with the time required for Problem 1. Which ran faster and by how much?
- 3) Compare the solutions for Problem 1 and Problem 2. Was the solution of Problem 2 as good as for Problem 1?

Extra Credit Problem

Revise your Python-MySQL-Gurobi program to incorporate a different transportation cost structure. Each trip still has a \$200 fixed cost and \$0.75 per mile traveled. But, in addition, if a trip is over 150 miles long then additional expense are incurred for driver lodging and meal allowance which totals \$250 because the driver will need to stay overnight in a hotel between the outbound and inbound legs of their trip. Implement an objective function with this conditional cost. Solve only Problem 1 with this new objective function. Submit this program to the separate Blackboard assignment that I have created for it.