Optimization
Final

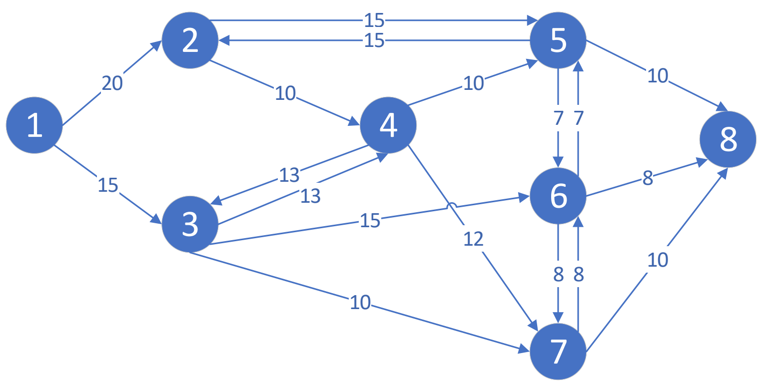
This is a Category A assignment.

You may not discuss, give, or receive assistance on this exam.

Fall 2018

Chocolate

Willy Wonka has a chocolate pipeline distribution network as shown below. Each node corresponds to a single storage tank, and the numbers on the edges represent flow capacities (per hour) of some unit of fluid chocolate. Note that flows are allowed in both directions between some, but not all, of the tanks. Willy would like to determine the maximum flow that can be sent from tank 1 to tank 8 per hour.



1. Formulate this problem in an Excel spreadsheet using Solver. Your deliverable should be called `p1.xlsx`.

2. As a sensitivity analysis, the company is also considering increasing the capacity of all arcs leading out of tank 1, and all arcs leading into tank 8, and it wants to know whether this will allow it to double the maximum flow per hour from tank 1 to tank 8. (Assume we multiply the arc capacities in question by a constant k, simultaneously.) Solve the original problem using an appropriate algorithm from the `optrees` or similar package in R, and ensure you achieve the same result as your Solver model. Then, perform the sensitivity analysis as requested. Is there a limit to increasing the expansion factor k? Your deliverable should be a simple R script called `p2.R`.

3. Extra Credit: Set up the simplex tableau (complete matrix in augmented form) as a simple matrix in R. Use the `rref()` function from the `pracma` package in R to solve. Deliverable is `p3_extra.R`.

Walrus World

A large retailer, Walrus World (WW), has dozens of regional distribution centers and thousands of 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 DC's, which are measured in the number of outgoing trailers per year that can be loaded from that facility. WW managers must also ensure that each retail store receives the number of trailers per week that are required to keep shelves stocked. Before the analysis can begin, some data wrangling must happen.

Part I

Provided for you are two tables in a database, `ww_dcs`, and `ww_stores`. When you examine these files, you will see that they are addresses of 46 dry goods distribution centers and 4535 stores, scraped from the web. Using Yahoo's Geocoding API (PlaceFinder), the addresses have been reconciled to *latitude* and *longitude* coordinates (in degrees).

4. Now that the web scraping and geocoding is complete, your first task is to pick up where the previous data scientist left off. You must write an R script to populate the table `ww_mileage`, consisting of all of the (dc,store) pairs. However, given the distances involved, the simple Euclidean distance will not do. Rather, you will employ the Haversine[1] distance formula, which takes into account the curvature of the Earth[2]. Though the function is simple to write, you can use the function `haversine()` from the R `pracma` package.

To receive credit, you must use R to read from the `ww_dcs` and `ww_stores` tables and write to the `ww_mileage` table in the database. Think of it as flexing your R muscles. (The astute analyst—you—should notice that we can solve this problem in a single SQL statement, with no R. But then you would not be flexing your R muscles, would you?) Leave your `ww_mileage` table populated. This deliverable is an R script `p4.R`.

Part II

To simplify things, you will solve this supply chain optimization for a subset of the DC's (10) and stores (1100) in the original problem. Data regarding the selected DC's and their capacities are contained in the `dc` table. Similarly, the `store` table specifies how many

[1] See `http://rosettacode.org/wiki/Haversine_formula` and Wikipedia: Haversine formula.

[2] After this rough optimization model is complete, a more accurate distance function can be used taking into account All-Pairs Shortest Paths using actual roads.

trailers per *year* are required by each store. The database also has a `mileage` table for the distance between each pairing of DC and store. The cost of each trip is a fixed \$200, plus \$0.75 per mile.

5. Formulate the problem using Solver, using just a few samples of rows (say, 4 each) from the `dc` and `store` tables. This is part of the prototyping process for your model. The deliverable is `p5.xlsx`.

6. Write a Python program that optimizes the same *prototype* model using Gurobi, and validate your solution by ensuring your results match your Solver model. Thus, you will need to use the same sample inputs as in your Solver model. This deliverable is `p6.py`.

7. Now for the real model. Write a Python program that optimizes this problem. To receive credit, you must dynamically retrieve the data from the database tables mentioned above to drive your model (`store`, `dc`, and `mileage`). After solving the model using Gurobi, write the results back to the `result` table. The deliverable is `p7.py`.

8. Use `mysqldump` or other export feature (e.g. within MySQL Bench) to create a dump of your database. If your username is `jsmith`, your dump file should be `jsmith.sql`. You can optionally compress the file as `jsmith_sql.zip`. Please substitute *your* WM user name as appropriate.

9. How would you 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. In addition, if a trip is over 150 miles long then an additional expense of \$250 is incurred for driver lodging and meal allowance—this is because the driver will need to stay overnight in a hotel between the outbound and inbound legs of their trip. *Describe* an objective function with this conditional cost, and any other changes to your model. Modify your Solver prototype to incorporate this new objective and deliver as `p9.xlsx`.

10. Submit a summary of your results—one for each problem—as `lastname.PDF`. If the question involved an optimization, here is where you include your optimal objective function value, and any commentary or business interpretation. If your solution for that problem is non-functional for any reason, please provide your perspective here. Also, if you made any modeling assumptions, included any tricky code, or just wish to say "hello, it's me", please include those comments here. All of your results should be posted to Blackboard in a .ZIP archive, by end of day, Dec 14, 2018. An extension of at most 1 day will be granted by permission.

Congratulations! You're now awesome. Go forthwith, in your newly bestowed awesomeness. Deliver, always, with excellence. Be a good person.