Solver Prototype to Gurobi Model

Dr. Aaron Koehl

In this assignment, we will continue to get some practice with linear optimization. In the videos, we developed a few linear programming models which you will now have an opportunity to implement in Gurobi and Python. You may use either Solver or OpenSolver (see syllabus) to develop your prototype. You will also be exposed to a *fundamental modeling step*: start with a known model that works, and extend it *iteratively* by testing after each modification. Then, if you run into an issue, it will be much easier to narrow down the source of the problem.

Related Reading: *Practical Management Science* Chapters 3 and 4

1 Assignment Overview

Your assignment is to translate some of the models we built in the videos, using Gurobi. You will complete this assignment first using Solver, then Python. Spreadsheet models are a little easier to see and more forgiving than their algebraic formulations. Thus, they are great prototyping tools, even when working on enormous problems in industry. Once you are sure that you have a working model, you can approach your translation to Gurobi with confidence. There is one additional problem (which we did not cover in the videos) that you will first have to solve using Solver, then translate to Gurobi.

1.1 Success Criteria

You are successful when you can show that your Gurobi model agrees with (returns the same result as) your Excel model for the same inputs.

1.2 Deliverables

Submit five files containing all of your work. The following parameters apply to this assignment, so be sure to review each of them before completing your work:

1. A single Excel Spreadsheet, `lastname-m3.xlsx`, with four tabs entitled "1, 2, 3, and 4" corresponding to the exercises below. Each tab should contain a solved model using Solver or Opensolver, and with the optimal solution showing.

2. Submit four Python scripts which use Gurobi, entitled `lastname-m3-schedule.py`, `lastname-m3-aggregate.py`, `lastname-m3-jones.py`, and `lastname-m3-blending.py`. The first three models are covered in the videos, and the fourth is from section 4.5 in *Practical Management Science*. The scripts must execute using Python 3.

3. Gurobi output is not suitable for a Business Analytics presentation by itself. For each model, you should write a formatted `print` statement which should be the last line printed. This should be a plain English sentence using the optimal solution. Do not hard-code the numbers/solution, but use the Python variables from the Gurobi model in your print statement, so that if the model inputs change then your printed output changes. Do not forget dollar signs, and round your financial figures to two decimal places as appropriate. Phrase your output as an analyst making a recommendation.

4. For the fourth problem (blending model), an algebraic formulation is not given to you. Try first writing it out on paper. Keep in mind that the constraint must be written in such a way that it is linear (as described on page 169 and the *Fundamental Insight* box on page 172).

5. For the fourth problem (after you have a working Python program), vary the selling price of gasoline in \$5 increments from \$40 to \$100 using a `for` loop, and for each selling price solve for the maximized revenue. (For each price use `m.update()` followed by `m.optimize()`). Use `matplotlib` to display a line plot of selling price along the horizontal axis, and revenue along the vertical axis. Do not forget a chart title and axis labels.

Exercises

1. The Lifeguard Scheduling Problem
2. The Multi-period (Aggregate Planning) Model
3. The Chuck Jones Financial Portfolio Planning Model
4. Chandler Blending

Last updated: July 12, 2019