SAS Optimization Challenge: Case Study

Title: HELP YOUR OPTIMIZATION PROFESSOR!!!!!

TIER 1 (of 5)

Tier 1: Modeling Details

The Tier 1 objective is to use the OPTMODEL procedure in SAS to choose the fifty (50) questions that produces **the most difficult exam possible** from the 154-question question bank, while ensuring that each *topic objective* contains the desired number of questions (i.e., provided in the **objective_questions.txt** data set).

For Tier 1, ignore the **Frenemy_Group** column.

Use the `create data` statement to generate one (1) output SAS data set containing only the fifty (50) questions that your model chose for the final exam. Include the following columns in the output data set: **Question**, **Topic**, **Objective**, and **Correct**. Questions that were not chosen should not appear in your output data set.

Additionally, report the average percentage correct (i.e., take the average of the **Correct** column) across all fifty (50) questions chosen by your model. This value can be calculated by any means necessary (e.g., in SAS, Excel, calculator, etc.).

To do this in SAS, copy and paste the following SQL procedure code at the end of your program, and replace *<insert output data set name>* with the name of your output data set that you created in the OPTMODEL procedure.

```
proc sql;
 select avg(Correct) as Avg_Correct
  from <insert output data set name>;
quit;
```

Tier 1: Getting Started

To get started, copy and paste the SAS® DATA step and datalines code from **question_bank.txt** and **objective_questions.txt** into your SAS® environment. Run the program to create the **work.question_bank** and **work.objective_questions** data sets, respectively. The two data sets are ready to be read into the OPTMODEL procedure.

Tier 1: Hints

The hints below are provided to help overcome relatively nuanced programming syntax and model formulation barriers that may arise in Tier 1.

I. Due to the hierarchical structure of the questions, the set of QUESTIONS should be indexed by the **Question**, **Topic**, and **Objective** columns to read the **work.question_bank** data set into the OPTMODEL procedure.

(e.g., `set <str,str,str> QUESTIONS;`)

II. To avoid confusion, consider using local dummy parameters `q`, `t`, and `o`, respectively.

III. Create an additional set to read the **work.objective_questions** data set into the OPTMODEL procedure. For consistency, consider naming the set OBJECTIVES.

IV. You can solve this as either a minimization problem (i.e., minimize exam easiness) or as a maximization problem (i.e., maximize exam difficulty).

V. If you choose to maximize exam difficulty, create a new parameter inside of the OPTMODEL procedure called difficulty, defined as (1 – **Correct**) for each question.

(e.g.,
```
num difficulty{<q,t,o> in QUESTIONS} = 1 -
    correct[q,t,o];
```
)

VI. The model should contain one constraint for each objective (i.e., *topic objective*). When summing across all questions, since both the OBJECTIVES set and the QUESTIONS set are indexed by objectives, use parenthesis ( ) on the local dummy parameter `o` in the QUESTIONS set to mask the implicit declaration.

(e.g., `sum{<q,t,(o)> in QUESTIONS}`)

VII. Use the colon operator to restrict the output table to contain only the fifty (50) questions chosen for the final exam.

VIII. Consider building your model on a subset of the questions, topics, and objectives first. Use the `expand` statement to unpack this smaller problem to ensure the model is performing correctly before rolling it out to all one hundred fifty-four (154) questions and sixteen (16) topic objectives.

Tier 1: Frequently Asked Questions

Q: I found a manual way to solve the Tier 1 problem quickly without using optimization. Should I bypass building the optimization model?

A: No. While it's a clever workaround, only solving the problem manually will burn you in subsequent tiers when the problem becomes more difficult. You could, however, solve Tier 1 using both optimization and non-optimization approaches to verify your results before moving onto Tier 2.

Q: I'm more familiar with the Excel solver. Can I use the Excel solver to solve this problem?

A: You can use whatever means necessary to solve the Tier 1 problem, so long as you also use the OPTMODEL procedure in SAS. See the Evaluation Criteria section in the Case Study Overview document.

When learning new technology, it can be helpful to first solve the problem using the technology you're currently familiar with as a starting point. If solving the Tier 1 problem first using the Excel solver helps you to formulate and solve the Tier 1 problem using the algebraic modeling language syntax of the OPTMODEL procedure, then you're encouraged to do it.

Q: Do I need to do any preprocessing of the two data sets before reading them into the OPTMODEL procedure?

A: No. The two data sets are ready to be read into the OPTMODEL procedure. See the Tier 1: Getting Started section for instructions.

Q: Do I need to do any postprocessing on the output data set containing the model results?

A: Yes, part of the submission guidelines for Tier 1 requires you to report the average percentage correct (i.e., the average of the **Correct** column) across all fifty chosen (50) questions in your output data set. To do this in SAS, use the SQL procedure code provided in the Tier 1: Modeling Details section above.

Q: Couldn't I just calculate every 50-question combination and filter the results?

A: No. That approach may work for smaller problems, but using the formula $nCr = \frac{n!}{r!(n-r)!}$, where n is the total number of questions in the set (i.e., 154), and r is the number of questions chosen from the set (i.e., 50), there are **9.86E40** different 50-question combinations that you'd need to calculate.

In other words, if you calculated 100 trillion combinations per second beginning the second that the universe was created, you still would not have calculated all 9.86E40 50-question combinations by the time your case study is due. By contrast, your optimization model should solve in less time than it takes to blink.

Q: I specified the solve statement (`solve;`) so that OPTMODEL would find and use the appropriate solver for this problem, but I saw the following note in the log. What does it mean?

```
NOTE: The problem has a decomposable structure with <xx> blocks. The
largest block covers <yy>% of the constraints in the problem.

The DECOMP option with METHOD=CONCOMP is recommended for solving
problems with this structure.
```

A: The OPTMODEL procedure recognized your problem correctly and used the default algorithm associated with the MILP solver. However, the OPTMODEL procedure has numerous non-default algorithms that can sometimes perform better than the default.

Based on the structure of the model, the OPTMODEL procedure recognized a non-default algorithm (i.e., the Decomposition Algorithm) that may solve your problem faster.

To use it, replace the original solve statement with: `solve with milp / decomp=(method=concomp);`

The pragmatic approach is to try both algorithms and use whichever one solves your model the fastest.