SAS Optimization Challenge: Case Study

Title: HELP YOUR OPTIMIZATION PROFESSOR!!!!!

TIER 4 (of 5)

Tier 4: Modeling Details

Oops! Your absentminded professor completely forgot about the practice exam! To help students prepare for the final exam, your professor always develops a comparable practice exam to ensure his students are ready for the final. The problem is that he completely forgot to tell you about this, and you can't read his mind!

Everything you've done up to this point has been for only one exam (i.e., the final exam, now referred to as the "Live" exam), but your professor needs two exams: a **Practice** exam, *and* a **Live** exam.

The model you built in Tier 3 is a good starting point. The exam generated in Tier 3 meets all of the professor's criteria for the Live exam. Those same requirements apply in Tier 4, but they now apply to both the Practice and Live exams.

The Tier 4 objective is to find the two ***most difficult***, ***balanced*** exams from the question bank, while ensuring the following requirements are satisfied ***for each exam***:

I. Each *topic objective* contains the correct number of questions for each exam.
II. No more than one question from each frenemy group is assigned to an exam.
III. The same question cannot appear on both exams. In other words, questions that appear on the Practice exam cannot appear on the Live exam, and vice versa.
    - Out of the 154 questions in the question bank, one hundred (100) total questions will be assigned in Tier 4:
        - Fifty (50) to the Practice exam
        - Fifty (50) to the Live exam

The eight (8) questions forced into the Tier 3 final exam must be forced into the Live exam in Tier 4. Consequently, to satisfy (III.) above, those eight (8) questions cannot appear on the Practice exam.

Balanced exams are exams with the same level of difficulty across all fifty (50) questions.

Recall that in Tiers 1-3 the *average percent correct* was calculated by taking the average of the **Correct** column across all fifty (50) exam questions chosen by the model. For the two exams to be balanced, the *average percent correct* across the fifty (50) questions in both the Practice and Live exams must be the same.

While countless balanced exams likely exist at varying difficulty levels, your objective in Tier 4 is to find the two ***most difficult***, *balanced* exams while controlling for the requirements stated above for each exam.

Use the `create data` statement to generate one (1) output SAS data set containing only the one hundred (100) questions that your model assigned to either the Practice or Live exam. Include the following columns in the output data set: **Exam, Question**, **Frenemy_Group**, **Topic**, **Objective**, **Force_Status**, and **Correct**. Questions that were not chosen should not appear in your output data set.

Additionally, report the *average percentage correct* for both exams (i.e., take the average of the **Correct** column) across all fifty (50) questions chosen by your model for each exam. To be balanced, these values must be identical, and 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 the output data set that you created in the OPTMODEL procedure.

```
proc sql;
 select distinct exam,
                 count(distinct Question) as Questions,
                 avg(Correct) as Avg_Correct
  from <insert output data set name>
   group by exam
    order by exam;
quit;
```

Tier 4: Getting Started

Copy and paste your Tier 3 solution (i.e., DATA step and datalines code creating the two data sets, the DATA step to create the **Force_Status** column, and OPTMODEL code) into a new programming editor to begin.

Modifications to the two data sets are not needed.

Tier 4: Hints

The hints below are provided to help overcome relatively nuanced programming syntax and model formulation barriers that may arise in Tier 4.

I. Create a new set called EXAMS, and explicitly declare the two elements `Practice` and `Live`. Since the elements in this set will never change, you will not be penalized in the **Data/Model Separability** section on the evaluation.

(e.g., `set EXAMS = /Practice Live/;`)

II. To avoid confusion, consider using local dummy parameter `e` for the set of EXAMS.

III. Revisit your current decision variable(s), for loop(s), and constraint(s) from your Tier 3 solution. Some, if not all, may need to be revised to accommodate the new EXAMS set.

For example, the decision variables declared using the `var` statement in Tiers 1-3 should've been indexed by the set of QUESTIONS to create one decision variable for each question. In Tier 4, those decision variables should be revised to now be indexed by both QUESTIONS and EXAMS.

IV. To explicitly declare an element from the EXAMS set in your model, use single quotes. This type of explicit declaration may be necessary in some, if not all, of the new Tier 4 constraints. When done correctly, the string will turn `purple`.

(e.g., `print (card({e in EXAMS: e = 'Practice'}));`)

V. Add a new conflict constraint for each question stating that each question can appear on at most one exam. This constraint will prevent exam questions from appearing on both exams.

VI. Your objective function is no longer to maximize exam difficulty or minimize exam easiness, whichever you chose in Tiers 1-3. Since the objective is to now generate balanced exams, the Tier 4 objective function is to minimize the absolute value of the difference in difficulty (or easiness) between the **Practice** and **Live** exams.

If your objective function in Tiers 1-3 was to *maximize exam difficulty*, your Tier 4 objective function is:

$$\min Z = |\text{Live exam difficulty} - \text{Practice exam difficulty}|$$

If your objective function in Tiers 1-3 was to *minimize exam easiness*, your Tier 4 objective function is:

$$\min Z = |\text{Practice exam easiness} - \text{Live exam easiness}|$$

For the new Tier 4 objective function, if the optimal objective value, Z, equals 0, then both exams are balanced. Otherwise, if Z is > 0, the two exams are unbalanced by the value of Z.

The problem with formulating the objective function this way is that they are both non-linear (i.e., due to the absolute value).

By definition, the mixed-integer linear programming solver can handle only linear objective functions, and the non-linear programming solver can handle only continuous decision variables, so neither solver is able to accommodate this model formulation.

For example, if you tried to use the mixed-integer linear programming solver (i.e., `solve with milp;`), you would receive the following error: `ERROR: The specified optimization technique does not allow nonlinear objectives.`

If you tried to use the non-linear programming solver to accommodate the non-linear objective (i.e., `solve with nlp;`), you would receive the following error: `ERROR: The NLP solver does not allow integer variables.`

Fortunately, both objective functions above can be reformulated as the linearized equivalent to absolute value. Once you reformulate the model to linearize absolute value in the objective function, the mixed-integer linear programming solver can easily accommodate the new Tier 4 requirements and solve the problem.

The question you need to find the answer to (and incorporate into your OPTMODEL program) is, "How do you linearize the absolute value of the difference between two positive values"?

The link below is one of many online resources that you may find helpful. Otherwise, happy Googling and ChatGPT'ng.

https://www.youtube.com/watch?v=oy3ffp_quSI

VII. Include a new constraint stating that the difficulty (easiness) of the **Live** exam must be $\geq$ ($\leq$) a user-defined cutoff value. Since numerous balanced exams exist at varying difficulty levels, this constraint allows you to find the ***most difficult*** of the balanced exams.

If your objective function in Tiers 1-3 was to *maximize exam difficulty*, this new Tier 4 constraint is:

$$\text{Live Exam Difficulty} \geq cutoff$$

Live Exam Difficulty is the sum of the **Difficulty** parameter across all questions assigned to the Live exam. The larger the Live Exam Difficulty value is, the more difficult the Live exam is, but the Live exam can only be so difficult. In other words, specifying a cutoff value that is too large will cause the problem becomes infeasible. For example, specifying a cutoff value of 1,000,000 will cause the problem will be infeasible because no fifty (50) question exam exists with a difficulty level of at least one million.

However, your Tier 3 objective value found a natural upper bound for the righthand side of this new inequality constraint (i.e., the cutoff value).

Recall that the Tier 3 objective was to maximize final (i.e., Live) exam difficulty, subject to the same conditions imposed on the Live exam in Tier 4. This means that Live Exam Difficulty in Tier 4 can't exceed your Tier 3 objective value.

$$\text{Live Exam Difficulty} \geq (\text{tolerance}) \times \text{Tier 3 Objective Value}$$

The tolerance parameter on the righthand side of the inequality ranges from 0 to 1, and when multiplied by your Tier 3 objective value, creates the cutoff value equaling a percentage of your Tier 3 objective value.

Your goal with this constraint is to determine what the tolerance value should be to generate the ***most difficult*** of the balanced exams. This will require running the model multiple times with different tolerance parameters. If the tolerance parameter is too high, the Practice and Live exams won't be balanced. If it's too low, the Practice and Live exams won't be the most difficult of the balanced exams. Try

numerous tolerance values until you've found the ***most difficult***, ***balanced*** exams possible.

Tuning the tolerance parameter should be **the very last thing you do** once your Tier 4 model is generating balanced Practice and Live exams of any difficulty level. Use 0 as a placeholder for the righthand side of the constraint until you're ready to begin manually tuning the tolerance value.

If your objective function in Tiers 1-3 was to *minimize exam easiness*, this new Tier 4 constraint is:

$$\text{Live Exam Easiness} \leq cutoff$$

Live Exam Easiness is the sum of the **Correct** parameter across all questions assigned to the Live exam. The smaller the Live Exam Easiness value is, the more difficult the Live exam is, but the Live exam can only be so difficult. In other words, if you specify a cutoff value that is too small, the problem becomes infeasible. For example, specifying a cutoff value of 3 will cause the problem will be infeasible because no fifty (50) question exam exists with a Live Exam Easiness level that low.

However, your Tier 3 objective value found a natural lower bound for the righthand side of this new inequality constraint (i.e., the cutoff value).

Recall that your Tier 3 objective was to minimize final (i.e., Live) exam easiness, subject to the same conditions imposed on the Live exam in Tier 4. This means that the Live Exam Easiness value in Tier 4 can't be less than your Tier 3 objective value.

$$\text{Live Exam Easiness} \leq (\text{tolerance}) \times \text{Tier 3 Objective Value}$$

The tolerance parameter on the righthand side of the inequality ranges from 1 to ∞, and can take fractional values (e.g., 4.0715).

When this tolerance parameter is multiplied by your Tier 3 objective value, it creates a multiplier of the lower bound.

Your goal with this constraint is to determine what the tolerance value should be to generate the ***most difficult*** (i.e., ***least easy***) of the balanced exams. This will require running the model multiple times with different tolerance parameters. If the tolerance parameter is too low (i.e., equal to 1), the Practice and Live exams won't be balanced. If it's too high, the Practice and Live exams won't be the most difficult possible exams. Try numerous tolerance values until you've found the ***most difficult***, ***balanced*** exams possible.

Tuning the tolerance value should be **the very last thing you do** once your Tier 4 model is generating balanced Practice and Live exams of any difficulty level. Use a sufficiently large number (e.g., 500) as a placeholder for the righthand side of the constraint until you're ready to begin tuning the tolerance values.