Chemical Engineering 4G03

Tutorial 7 Practice Problems

Discrete Programming and Formulation Practice

Here for your own benefit and practice (best to do it individually)
Recommended completion: Week 08.
Grading: 0% (Practice for assignments and tests)

Problem 1 – Bread, Math, and Beyond

You are the manager of a small bakery commonly used as a study area for first year calculus called "Bread, Math and Beyond." You are considering the upcoming week in which you want to determine how many batches of your products to make based on some prior demand data. For the sake of argument, let's say you can make six products: bread (BRD), cookies (CKS), Rolls (ROL), muffins (MUF), crescents (CRE), and scones (SCO). The demographic data that you anticipate (maximum demands and profit per batch) for the coming week are reported in the table below. Moreover, the mixing time (MIX), baking time (BAK) and packing time (PAK) required for each batch is reported in the same table.

Product	Max Demand (Batch)	Profit ($/Batch)	Mixing (min/Batch)	Baking (min/Batch)	Packing (min/Batch)
BRD	20	12	25	14	6
CKS	60	16	28	26	9
ROL	25	13	16	16	7
MUF	40	19	35	10	8
CRE	75	9	16	24	4
SCO	35	20	22	21	9

There are several industrial stand mixers ovens available. You may assume that all tasks may be done **at the same time.** Based on the capacities of these units you have:

- **4800 minutes** of mixing time available per week.
- **4800 minutes** of baking time per week.
- You are the only employee working in the store and thus you have **1200 minutes of packing time** available per week.

Questions

1. Formulate this as an LP and enter it into GAMS as a starting point. We are going to introduce some binary variables as we go that will affect our solution. Run your code and **verify** that the optimal profit is $2644 per week.

2. OK, let's start to add some yes/no decisions in here. First, consider the case where, to avoid complications and to limit your orders, you are limited to producing only 4 different products per week. Add the constraint and variables that will make this happen. Add the binary variables and constraints to your GAMS code by declaring the binary as `BINARY VARIABLE`. You must change

the solve statement to solve using the `MIP` solver (NOT the `LP` solver). The MIP default is CPLEX, so you will not need to change that option. Before running your code, ask yourself: is the profit *guaranteed* to be worse, better, or the same? Verify your result by running the code.

3. Repeat the above but do it for 3 maximum products and 2 maximum products. Do the active constraints change significantly as you reduce the number of products you are making? What about the combination of products you are choosing?

4. In this scenario, we can make anything we want, BUT we are obliged to make 20 batches of cookies we promised to a local retailer (Café Oranje). Reformulate your program to account for this new constraint and re-solve the problem.

5. Now consider the case where we can make anything we want again with no cookies required. HOWEVER, there is a fixed charge for making any one product of $300 (so $300 per product we choose to make in ANY quantity), which is the delivery charge for the necessary ingredients (also accounts for you investing your own time to unload and store all the ingredients). Reformulate your problem again to account for this new cost. Re-solve it using GAMS to report the new optimum.

6. Return to the base case again. Argue convincingly that if I were to impose on you the requirement "if you make any of one product, you must make at least 10 batches of it", the optimum will either go down or stay the same in the best possible case. This would be true if, for instance, ingredients come in minimum shipments of 10 batches-worth.

7. Re-solve the problem with this added constraint to prove your point.

8. Re-formulate and re-solve the optimization so that if I want to make any one product, I am obliged to make 30 batches. You may adjust the demand data such that any product with a demand < 30 can be equal to 30 instead. Comment on your findings.

Problem 2 – Optimization… IN SPAAAAAAACE!

The US space agency known as the National Aeronautics and Space Adventuring Helionauts (NASAH) typically has to face decisions around budgeting long-term missions and projects. For example, consider the case where they have funding available for a series of projects available over a five-stage period of time. Each project requires a certain amount of money in each stage, and each stage has a well-defined budget from the government. In the **table below** is a list of prospective projects and their anticipated cost at each stage of the budgeting plan. Furthermore, in the table is the expected "benefit" (an aggregate score based on likelihood of success and information gathered as a result of the mission) of each project and a list of projects that depend on or are mutually exclusive to it. Note that if you choose a project in the table below you **must pay for all costs in each stage for that project**.

This type of problem can be defined as a "multi-dimensional knapsack" problem in which we are choosing which projects to take depending on capacity (*i.e.* budget) constraints for multiple stages. It is our objective to allocate research funds that will maximize the benefit of NASAH's funding over the next five stages.

		Budget Requirement ($billions)							
i	Mission	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5	Value	Not With	Depends On
1	Comms Satellite	6	-	-	-	-	200	-	-
2	Orbital Microwave	2	3	-	-	-	3	-	-
3	IO Lander	3	5	-	-	-	20	-	-
4	Uranus Orbiter 2030	-	-	-	-	10	50	5	3
5	Uranus Orbiter 2040	-	5	8	-	-	70	4	3
6	Jupiter Probe	-	-	1	8	4	20	-	3
7	Saturn Probe	1	8	-	-	-	5	-	3
8	Infrared Imaging	-	-	-	5	-	10	11	-
9	Ground-Based SETI	4	5	-	-	-	200	13	-
10	Orbital Structures	-	8	4	-	-	150	-	-
11	Color Imaging	-	-	2	7	-	18	8	2
12	Medical Technology	5	7	-	-	-	8	-	-
13	Geosynchronous SETI	-	4	5	3	3	185	9	-
14	Jetpod Shielding	-	-	3	3	1	50	-	12
Budget		10	12	14	14	14			

Questions

1. Define the variables for this problem, clearly indicating if they are continuous or binary.

2. Define the objective for this problem. You may write this out in terms of hard-coded numbers or define new parameters as you see fit.

3. Define the constraint(s) that correspond(s) to the available budget for each stage in each for this problem. Try to be as concise as possible.

4. Code your formulation *so far* (that is, DO NOT include the mutual exclusivity or dependence constraints) in GAMS and find the solution.

5. Define the mutual exclusivity constraint for this problem.

6. Add the mutual exclusivity constraints to GAMS and re-run the problem. Comment on the results and how adding the mutual exclusivity constraints affect your solution.

7. Define the dependency constraints for this problem.

8. Add the dependency constraints to the **base case** in GAMS (that is, don't include the mutual exclusivity constraints) and re-run the problem. Comment on the results and how adding the dependence constraints affect your solution.

9. Re-run your GAMS code with *both* of the mutual/dependence constraints included. Comment on *this* solution.

10. Feel free to play around with costs, budgets and mutual exclusivities to see how they affect the problem!