

Homework 5

Linear Programming

Assigned: Fri Apr 1

Due: Fri Apr 8 at 4 pm

Go to www.ampl.com, click on **Try AMPL!**, accept the conditions, select **diet.mod** and **diet.dat** from the list of possible **model files** and **data files**, and click **Submit**. More in a moment, but first let me explain...

The **diet problem** is a classical example of LP. It's appealing not because it is a practical example, but because it is easy to understand, fun, and it's not hard to see how more practical problems might be expressed in such a framework. The idea of the diet problem is that a person is choosing a diet from a set of possible foods, with constraints on the nutrients included in the diet, as well as on the amounts of each food that can be consumed. The objective is to minimize the cost. In summary, the diet problem can be written in the form

$$\begin{array}{ll} \min & c^T x \\ \text{subject to} & g \leq Ax \leq h, \quad \ell \leq x \leq u \end{array}$$

Here the (i, j) entry of A , say a_{ij} , is the known amount of nutrient i in food j , and x_j is the unknown variable: the amount of food j to be in the diet. This must be between ℓ_j and u_j . On the other hand, summing $a_{ij}x_j$ tells you the total amount of nutrient i in the diet; this must be between g_i and h_i .

Going back to the web page: after selecting **diet.mod** and **diet.dat** and clicking **Submit**, carefully study the information in the window that is displayed. You see that a bunch of parameters (cost, amt, and so on) are declared. The **cost** vector is what we called c above, and the **amt** matrix is A . The quantity **amt**{NUTR,FOOD}, or in matrix notation a_{ij} , is the amount of nutrient NUTR in food FOOD. The other parameters are lower and upper bounds on the amount allowed of each food, and lower and upper bounds on the required total nutrients. Scrolling down, you see that **data** values are assigned to these parameters below (these come from the **diet.dat** file). Before assigning the numerical values, the NUTR and FOOD index sets are defined (NUTR: Vitamins A, B1, B2 and C; FOOD: Beef, Chicken, Fish, Ham, Macaroni and Cheese (!), Meat Loaf (!), Spaghetti and Turkey). Don't go any further until you understand the notation.

Now, turn your attention to the **Solver** menu. This selects between various software packages that can be used to solve the LP that is defined in the window. The default is MINOS, a top-quality Fortran package from Stanford written by Michael Saunders. Not all these solvers are actually relevant for solving LP's!

Using Minos, click **Send**. This solves the LP and you will see that the optimal solution is a diet consisting entirely of Macaroni and Cheese!

Questions to answer:

1. Obviously, all the variables are on their lower bounds of 0 except MCH. Thus, they are all active (nonbasic). But what about the nutrient constraints? Which of these are on their lower or upper bounds? In other words, which of these constraints are active? To answer this you either have to do some computation (e.g., by pasting the data into Matlab or Excel), or (better) figure out how to display what you want in AMPL by adding to the display directive in the middle window.
2. In the light of your answer to the previous question, is the optimal point displayed a *basic* optimal point? Since the formulation is not in standard form, you have to redefine what we mean by basic. It is not necessary to convert to standard form; you can argue geometrically.
3. A second diet data file **diet2.dat** attempts to define a more healthy diet. Select this instead of **diet.dat** (if your Back button doesn't work, you can start over; you may want to bookmark the page). Compare this data file with the first and observe the differences. Now click **Send** to solve this new problem. You see the LP is infeasible - too many constraints have been added. Relax some of the constraints in the new problem sufficiently (using a trial and error process) until you have an LP which is feasible and for which the solution is more interesting than the first problem. Print the web page, making sure both input and output are visible, and explain whether the optimal point displayed is a basic optimal point, arguing as earlier.
4. Get creative and introduce some new foods. You don't need to use real data. While you are at it, you may as well overhaul the choice of foods completely, using a less American (and more healthy?) cultural norm! The point of this question is to get you to play with the model, being careful to respect the AMPL syntax, and come up with a more

interesting diet. Print the resulting web page, making sure that all the input and output is visible. Again, explain whether the optimal point is basic.

5. Let's now simplify the LP to

$$\begin{array}{ll} \min & c^T x \\ \text{subject to} & Ax \geq b, \quad x \geq 0 \end{array}$$

This is still not standard form, since we have $Ax \geq b$, not $Ax = b$. Introduce slack variables to transform it to standard form, derive the dual LP (since we know what this is for standard form), and then simplify it if possible. The final dual LP should have a form that shows a beautiful symmetry with the original primal LP. You might wonder why we don't use *this* as standard form; the answer is that the simplex method is fundamentally suited for the version with $Ax = b$.

6. As is almost always the case, there is an economic interpretation for the dual LP. In this case, the idea is that the dual LP expresses the goal of a pharmacist who is selling pills that provide the necessary vitamins to customers, so they can get the nutrients they need without food (yuck, but we need suspension of disbelief here). Explain the details, including the meaning of the dual variables and the constraints and why it makes sense economically that the primal and dual have the same optimal value. This requires some thought and some detailed explanation; work it out yourself if you can, but if you are stuck, look for help in the library or on the web, being sure to acknowledge your source, if any.
7. Finally, returning to the form of the diet problem given on the first page, introduce slack variables and derive the dual LP, simplifying it as much as you can.

AMPL stands for A Mathematical Programming Language. There are two copies of the AMPL book in the CIMS library. I don't think you need to consult this, but in case you wish to do so, I am putting them on reserve. There is also one more (in fact the latest version) at Bobst which is not on reserve and so available to the first enterprising person who wants to take it out!