GLPK Case Study 2 - 60016 Operations Research

A software development company wants to publish a mobile app to automatically generate grocery shopping lists. The software will use a web service to download the nutritional values of the foods available at the closest grocery shop, determined according to the location of the user, and then will produce a shopping list that maximizes the food calories that can be purchased for a given budget. The head of development has decided that the software will be interfaced to GLPK to generate the shopping recommendations. You are therefore given a sample GLPK data file (see Listing "food.dat") for testing purposes and you need to define the GMPL model that will be used to generate the shopping lists. In its basic version, the linear program has to express the following constraints:

- The shopping list specifies the amount in grams of each food to be purchased;
- The amounts of ingredients in the shopping list are at minimum the values specified in the nutrional requirements vector NR;
- The ingredients in 100 grams of food are specified by the FO matrix;
- The cost of each gram of food is specified by the CO vector;
- The maximum cost of the shopping list equals the Budget parameter;
- The objective function attempts to maximise the total number of calories in the shopping list;

(a) Write the GMPL linear program for the linear problem and find its optimal solution with glpsol;

(b) Using glpsol's output file, determine which variables are basic and which ones are nonbasic. Is the number of basic variables equal to the number of rows?

(c) In the optimal solution, verify that the reduced cost of Carrot in glpsol's output file corresponds to the value predicted by the theoretical formula.

Listing 1: food.dat

```
data;

param Budget := 10; /* pound sterlings */

set Foods := Apple Orange Oil Carrot Milk Chocolate Potato Spinach ;
set Ingredients := Calories Fats Carbs Proteins ;

param NR :=
Calories         2100     /* calories */
Fats             100      /* grams */
Carbs            225      /* grams */
Proteins         175 ;    /* grams */

/* ingredients per 100 grams of food */
param FO : Apple Orange Oil  Carrot Milk Chocolate Potato  Spinach :=
Calories    52     33     884  41     42     546      77       23
Fats        0.2    0.3    100  0.2     1      31      0.1      0.4
Carbs       14     8      0    10      5      61      17       3.6
Proteins    0.3    0.7    0    0.9    3.4    4.9       2       2.9  ;

param CO := /* cost per gram (pound sterlings) */
Apple            0.0065
Orange           0.0020
Oil              0.0330
Carrot           0.0070
Milk             0.0029
Chocolate        0.0140
Potato           0.0005
Spinach          0.0030  ;

end;
```

Solution

(a) See Listing "foods.mod". Since the data section is not included in the model file, glpsol needs to be called using the syntax: glpsol -m foods.mod -d food.dat -o foods.out . For the given dataset, the optimal objective function value is 11804.28 calories, and the optimal solution is a shopping list including only 14.35kg of potatoes and 85.65g of oil.

Listing 2: foods.mod

```
# food.mod
set Foods;
set Ingredients;

param CO {i in Foods}; /* cost per unit of food */
param NR {j in Ingredients}; /* nutritional requirement intake */
param FO {j in Ingredients, i in Foods}; /* units of ingredient j in food i */
param Budget;

# DECISION VARIABLES
var x {i in Foods}, >= 0; /* grams of food i in shopping list */

# LINEAR PROGRAM
maximize cal : sum {i in Foods} (FO["Calories",i] / 100) * x[i];

s.t.
MaxCost: sum {i in Foods} CO[i] * x[i] <= Budget;
NutrReq {j in Ingredients} : sum {i in Foods} (FO[j,i] / 100) * x[i] >= NR[j];

# SOLVE LINEAR PROGRAM
solve;

end;
```

(b) Looking at the columns table in the food.out file, we see that Potato and Oil are basic (B value in the St column), whereas all the other variables listed are non-basic. The number of basic variables (i.e., 1) is not equal to the number of rows which is 5 (note that GLPK includes the objective in the row count, since z is part of the basic representation). This is because we did not provide the program in standard form and GLPK by default does not list the slack/excess variables. Indeed, if we provide the standardized program, it becomes evident that also the slack variables for the nutritional requirement constraints are basic, whereas the excess variable for the budget constraint is non-basic. This can be seen by generating the output file for the Listing "food-std.mod"

Listing 3: food- std.mod

```
# food.mod
set Foods;
set Ingredients;

param CO {i in Foods}; /* cost per unit of food */
param NR {j in Ingredients}; /* nutritional requirement intake */
param FO {j in Ingredients, i in Foods}; /* units of ingredient j in food i */
param Budget;

# DECISION VARIABLES
var x {i in Foods}, >= 0; /* grams of food i in shopping list */
var s1 , >= 0; /* slack variable for MaxCost */
var s2 {j in Ingredients}, >= 0; /* excess variables for NutrReq */

# LINEAR PROGRAM
minimize cal : sum {i in Foods} - (FO["Calories",i] / 100) * x[i];

s.t.
MaxCost: sum {i in Foods} CO[i] * x[i] + s1 = Budget;
NutrReq {j in Ingredients} : sum {i in Foods} (FO[j,i] / 100) * x[i] - s2[j] = NR[j];

# SOLVE LINEAR PROGRAM
solve;

end;
```

(c) Looking again at the columns table in the food.out file, we see that the reduced cost of Carrot is -10.9094. Since this is a maximization problem not in standard form, we expect the theoretical formula to predict 10.9094, given that the standard form involves a minimization instead of a maximization. In standard form, the problem has coefficient matrix

$$A = \begin{pmatrix} .0065 & .0020 & .0330 & .0070 & .0029 & .0140 & .0005 & .0030 & 1 & 0 & 0 & 0 & 0 \\ 52 & 33 & 884 & 41 & 42 & 546 & 77 & 23 & 0 & 1 & 0 & 0 & 0 \\ 0.2 & 0.3 & 100 & 0.2 & 1 & 31 & 0.1 & 0.4 & 0 & 0 & 1 & 0 & 0 \\ 14 & 8 & 0 & 10 & 5 & 61 & 17 & 3.6 & 0 & 0 & 0 & 1 & 0 \\ 0.3 & 0.7 & 0 & 0.9 & 3.4 & 4.9 & 2 & 2.9 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

We know that at optimality Oil (column 3) and Potato (column 7) are basic together with the slack variables for the nutritional requirements constraints on Calories, Carbs, and Proteins. Therefore, the index set will be $I = [7, 9, 10, 11, 12]$. This implies

$$B = \begin{pmatrix} .0330 & .0005 & 0 & 0 & 0 \\ 884 & 77 & 1 & 0 & 0 \\ 100 & 0.1 & 0 & 0 & 0 \\ 0 & 17 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 & 1 \end{pmatrix}$$

$$N = \begin{pmatrix} .0065 & .0020 & .0070 & .0029 & .0140 & .0030 & 1 & 0 \\ 52 & 33 & 41 & 42 & 546 & 23 & 0 & 0 \\ 0.2 & 0.3 & 0.2 & 1 & 31 & 0.4 & 0 & 1 \\ 14 & 8 & 10 & 5 & 61 & 3.6 & 0 & 0 \\ 0.3 & 0.7 & 0.9 & 3.4 & 4.9 & 2.9 & 0 & 0 \end{pmatrix}$$

$$c_B^T = [-8.84, -0.77, 0, 0, 0]$$

$$c_N^T = [-0.52, -0.33, -0.41, -0.42, -5.46, -0.23, 0, 0]$$

and it can be verified that the entry for Carrot in $c_N^T - c_B^T B^{-1} N$, i.e., column 3, equals 10.9094.