Lesson 11. Resource Allocation Models, Revisited

1 Writing optimization models with symbolic input parameters

Problem 1. Farmer Jones decides to supplement her income by baking and selling two types of cakes, chocolate and vanilla. Each chocolate cake sold gives a profit of \$3, and the profit on each vanilla cake sold is \$4. Each chocolate cake uses 4 eggs and 4 pounds of flour, while each vanilla cake uses 2 eggs and 6 pounds of flour. Farmer Jones has 32 eggs and 48 pounds of flour available. Assume all cakes baked are sold, and fractional cakes are OK. Write a linear program that determines how many of each type of cake should Farmer Jones bake in order to maximize her profit.

Recall that the linear program we wrote for this problem is

C = number of chocolate cakes to bake
V = number of vanilla cakes to bake

$$\begin{array}{lll} \text{maximize} & 3C + 4V & \text{(total profit)} \\ \text{subject to} & 4C + 2V \leq 32 & \text{(eggs available)} \\ & 4C + 6V \leq 48 & \text{(flour available)} \\ & C \geq 0,\ V \geq 0 & \end{array}$$

Problem 2. Farmer Jones decides to supplement her income by baking and selling cakes. Let K be the set of cake types that she sells. Each cake k sold yields a profit of p_k, for all $k \in K$. Each cake type requires a certain mixture of ingredients. Let I be the set of ingredients that are used. Each type k cake requires a_{ik} units of ingredient i, for all $i \in I$ and $k \in K$. Farmer Jones has b_i units of ingredient i available, for all $i \in I$. Assume all cakes baked are sold, and fractional cakes are OK. Write a linear program that determines how many of each type of cake should Farmer Jones bake in order to maximize his profit.

- Recall that **input parameters** are quantities that are given and fixed
- What are the input parameters in Problem 2?

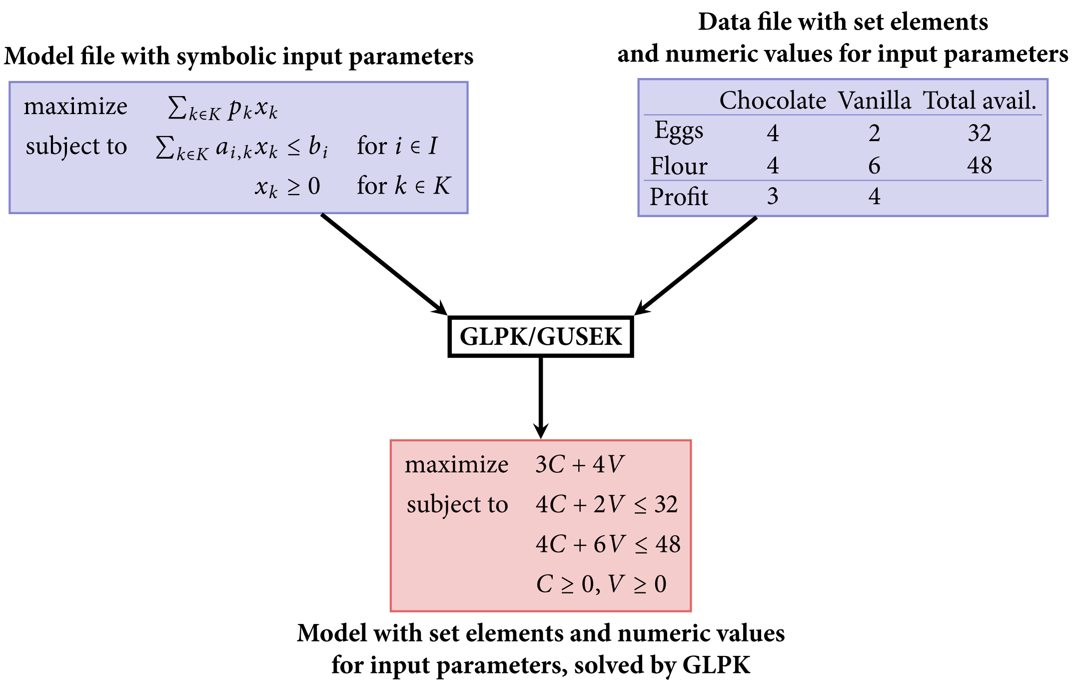
- How do these input parameters relate to the those given in Problem 1?

- Write a linear program for Problem 2, using summation notation and for statements.

- This model has **symbolic input parameters**
  - "Placeholders" for actual set elements and numerical values
- This model is valid for any problem of the same structure
  - Just need to specify actual set elements and numerical values for the symbolic input parameters
  - e.g. Specify elements for K and I; numerical values for p_k for $k \in K$, b_i for $i \in I$, and a_{ik} for $i \in I$ and $k \in K$

2 Sets, summations, for statements, and symbolic input parameters in GMPL

- How do we use sets, summation notation, for statements, and symbolic input parameters in GMPL?



- **GMPL model file** (`farmerjones.mod`)

```
## Input parameters ##
set K;                    # set of cake types
set I;                    # set of ingredients
param p{k in K};          # p[k] = profit for cake type k
param b{i in I};          # b[i] = amount of ingredient i available
param a{i in I, k in K};  # a[i,k] = amount of ingredient i used in 1 type k cake

## Decision variables and variable bounds ##
var x{k in K} >= 0;       # x[k] = number of type k cakes to produce

## Objective function ##
# Maximize total profit
maximize total_profit:
  sum{k in K} p[k] * x[k];

## General constraints ##
# Amount of ingredient i used <= amount of ingredient i available
subject to ingredient_avail{i in I}:
  sum{k in K} a[i,k] * x[k] <= b[i];

end;
```

- **GMPL data file** for the Problem 1 (`farmerjones-original.dat`)

```
# Input parameters for the original Farmer Jones problem in Lesson 11

# Set of cake types
set K := Chocolate Vanilla;

# Set of ingredients
set I := Eggs Flour;

# p[k] = profit for cake type k
param p :=
  Chocolate  3
  Vanilla    4;

# b[i] = amount of ingredient i available
param b :=
  Eggs        32
  Flour       48;

# a[i,k] = amount of ingredient i used in 1 type k cake
# rows correspond to i, columns correspond to k
param a:
              Chocolate   Vanilla :=
Eggs          4           2
Flour         4           6;

end;
```

- Running the model and data file in combination in GUSEK:
  - Make sure `farmerjones.mod` is the only model file open
  - Swtich to `farmerjones-original.dat`
  - Select Tools » Set as Default .dat File
  - Switch to `farmerjones.mod`
  - Make sure Tools » Generate Output File on Go is checked
  - Select Tools » Go
  - You can check if the model and data combine in the way you expect by selecting Tools » Build Cplex LP
    - Note that in a Cplex LP file, variables are assumed to be nonnegative unless otherwise specified
    - Do not follow this practice! Always specify nonnegativity constraints if necessary!
- Multiple summations: you can write

$$\sum_{i \in I} \sum_{j \in J} c_{i,j} x_{i,j}$$

  in GMPL like this:

```
sum{i in I, j in J} c[i,j] * x[i,j]
```

- Other examples of iterating over multiple sets:

```
var x{i in I, j in J};
subject to constraint_name{k in K, l in L}:  x[k] <= b[l];
```

Problem 3. Farmer Jones's cake business has been quite successful! With some new recipes in hand, she is trying to determine how to expand her cake offerings. Farmer Jones can now bake and sell 3 types of cakes: chocolate, vanilla, red velvet. Each cake requires varying amounts of 4 ingredients: prep time, baking time, eggs, flour. In particular, the amount of each ingredient needed in each type of cake is given below:

	Chocolate	Vanilla	Red Velvet
prep time	30	20	50
baking time	25	40	35
eggs	3	2	4
flour	4	4	5

Each chocolate cake generates a profit of \$4, vanilla \$5, and red velvet \$5. Farmer Jones has 240 minutes of prep time, 280 minutes of baking time, 50 eggs, and 40 pounds of flour available. Write a data file `farmerjones-new.dat` that accompanies the model file `farmerjones.mod` to solve Farmer Jones's new problem. Solve the linear program.