Dept. of Computer Science and Engineering,

CS69011: Computing Lab 1

Assignment 4

Maximum Marks: 30

In this assignment, we will solve some real-world problems using
Linear Programming (https://en.wikipedia.org/wiki/Linear_programming) and
Integer Programming (https://en.wikipedia.org/wiki/Integer_programming).
We are going to use the opensource library GLPK (GNU Linear Programming Kit, https://www.gnu.org/software/glpk/) for solving the optimization problem.
In this assignment we are going to solve the
Optimal Transport problem (https://en.wikipedia.org/wiki/Transportation_theory_(mathematics)) and
Facility location problem (https://en.wikipedia.org/wiki/Facility_location_problem).

Task 1

Consider that an army has located its units in n locations and has to supply soldiers to m battlegrounds. The cost of supplying one soldier to from unit i to battleground j, is c_{ij}. Also, there is a demand of at least d_j soldiers in battleground j, and there is an upper bound of u_i on the total number of soldiers who can be accommodated at unit location i. The task is to find the optimal fraction of demand for soldiers d_j to be met by the unit location i, denoted as x_{ij}. This can be obtained by solving the linear program:

$$\min_{\mathrm{x_{ij}}} \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} * d_j * x_{ij}$$

$$sub.to. \sum_{j=1}^{m} d_j x_{ij} \le u_i \quad \forall i = 1, \dots, n$$

$$\sum_{i=1}^{n} x_{ij} \ge 1 \quad \forall j = 1, \dots, m$$

$$x_{ij} \in [0,1] \quad \forall i, j$$

Here, the objective function measures total cost of transporting all soldiers. Note that, $d_j * x_{ij}$ are the number of soldiers supplied from location i to battleground j. The first set of constraints ensures that not more than u_i soldiers are supplied from location i, and the second set of constraints ensure that at least d_j soldiers are supplied to battleground j.

Input:

The input file format is:

<value of n> <value of m>

<vector of u_i (n numbers in one line)>

<vector of d_j (m numbers in one line)>

<first row of cost matrix c_{ij} (m numbers in one line)>

…

<last (nth) row of cost matrix c_{ij} (m numbers in one line)>

Output:

Print input data: values of n and m, u vector, d vector, and c matrix.

Print the matrix of optimal allocation of soldiers from unit location I to battleground j: $d_j * x_{ij}$

Task 2:

Consider the problem of locating army units in at most n locations (facility points) for servicing the needs of $m$battle grounds (demand points). Each facility point i has a cost of c_{ij} of supplying the demand point j, this could be the cost of transporting one soldier to from unit location i to battleground j. Each battleground j has a demand for d_j soldiers, assumed to be known. Moreover, each facility i has an initial fixed cost of f_i of setting up, and an upper bound u_i on the demand for number of soldiers which can be accommodated. Let us say we want to open k of the n facilities. The problem is to find out the optimal fraction x_{ij} of the soldiers d_j which will be supplied by facility location i to battleground j. Additionally, we must find out which of the n locations should be used to set up unit facilities, maximum number of them being k. This can be solved using a mixed integer linear programming problem. Let y_i be a binary integer variable denoting whether facility location i should be opened ($y_i = 1$) or not ($y_i = 0$). The optimization problem becomes:

$$\min_{x_{ij}} \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} * d_j * x_{ij} + \sum_{i=1}^{n} f_i * y_i$$

$$sub.to. \sum_{j=1}^{m} d_j x_{ij} \le u_i y_i \quad \forall i = 1, \dots, n$$

$$\sum_{i=1}^{n} x_{ij} \ge 1 \quad \forall j = 1, \dots, m$$

$$x_{ij} \in [0,1] \quad \forall i, j$$

$$y_i \in \{0,1\} \quad \forall i$$

Here, the objective function measures total cost of transporting all soldiers (first term) and the total setup cost of all open facilities (second term). Note that, $d_j * x_{ij}$ are the number of soldiers supplied from location i to battleground j. If location i is not open ($y_i = 0$) the upper bound is 0, otherwise it is u_i. The first set of constraints ensures that not more than u_i soldiers are supplied from location i, and the second set of constraints ensure that at least d_j soldiers are supplied to battleground j. Here, y_i are integral binary variables.

Input:

The input file format is:

<value of n> <value of m>

<vector of u_i (n numbers in one line)>

<vector of f_i (n numbers in one line)>

<vector of d_j (m numbers in one line)>

<first row of cost matrix c_{ij} (m numbers in one line)>

…

<last (nth) row of cost matrix c_{ij} (m numbers in one line)>

Output:

Print input data: values of n and m, u vector, f vector, d vector, and c matrix.

Print the list of opened facilities: y vector.

Print the matrix of optimal allocation of soldiers from unit location I to battleground j: $d_j * x_{ij}$

GLPK Guide:

Get yourself introduced to the GLPK API. If you want to install it in your personal machine (like laptop), you need to download the package from the standard repositories (http://ftp.gnu.org/gnu/glpk/) and compiling it from source. For Ubuntu, pre-built binaries can be installed using:

```
$ sudo apt install glpk-utils libglpk-dev glpk-doc
```

Read the user's manual from (https://cse.iitkgp.ac.in/~abhij/course/lab/CompLab-I/Autumn19/glpk.pdf). Include the following directive in your program.

```
#include <glpk.h>
```

Compile your code with the following flags.

```
gcc -Wall glpkdemo.c -lglpk -lm
```

GLPK supports both real-valued linear programming and mixed-integer optimization. The basic API calls are `glp_simplex` and `glp_intopt`. The integer optimizer needs an initial solution. You can start with the output of the simplex solver.