Combinatorial Optimization
Lab No. 8:
Maximum Network Flows

Industrial Informatics Department
Czech Technical University in Prague
https://industrialinformatics.fel.cvut.cz/

April 15, 2024

Abstract

In this lab, we familiarize ourselves with Network Flows. We will learn how to solve Maximum Network Flow Problem and how to model practical problems as an instance of this problem.

1 Ford-Fulkerson Algorithm

Ford-Fulkerson is the basic algorithm for solving Maximum Network Flow Problem. It relies on the theorem that states: a flow is maximal if and only if there is no $s - t$ augmenting path. Existence of such a path can be checked by running so-called *labeling procedure* (see lectures).

The disadvantage of the Ford-Fulkerson algorithm is its running time $\mathcal{O}(mF)$, where F is the size of maximal flow. Unfortunately, this is not polynomial in the length of the input.

The complete pseudocode of Ford-Fulkerson algorithm is given in Algorithm 1.

Algorithm 1 Ford-Fulkerson algorithm

```
Require: initial feasible s − t flow f in (G, l, u)
Ensure: maximal feasible s − t flow in (G, l, u)
  function FORD-FULKERSON(G, l, u, f, s, t)
      while True do
          δ, P ← labeling(G, l, u, f, s, t)
          if δ > 0 then
              for all e ∈ P do
                  if e is a forward arc in G then
                      f(e) ← f(e) + δ
                  else
                      f(e) ← f(e) − δ
                  end if
              end for
          else
              break
          end if
      end while
      return f
  end function
```

Protip: Instead of choosing an arbitrary augmenting path from labeling procedure, always pick the shortest one in terms of edges. This gives you asymptotically faster algorithm known as Edmons-Karp algorithm that runs in $\mathcal{O}(nm^2)$.

2 Homework: Survey Design

The Coconut Craftworks, Co. manufactures a set of various products $\mathcal{P}$. The company has selected a set of their customers $\mathcal{C}$ to participate in a survey about its products. Since the company keeps track on the past orders made by customers they know for each customer $c_i \in \mathcal{C}$ what subset of products $P_i \subseteq \mathcal{P}$ customers bought in the past.

Each customer $c_i \in \mathcal{C}$ has expressed the interest to review at least l_i products (*to get a discount for future purchases*) but at most u_i of them (*to not to get overloaded*). The marketing head of *The Coconut Craftworks, Co.* requires to obtain at least v_j reviews for each product $j \in \mathcal{P}$ in order to have a good estimate of how much the products are likable by customers.

Your tasks is to design a survey — distribute product surveys among customers, such that

- each customer is reviewing products that he/she knows;
- each customer gives at most one review per product;
- the total number of reviews each customer i provides lies in interval $[l_i, u_i]$;
- each product j is reviewed by at least v_j customers;
- the total number of reviews is maximized.

Homework assignment: Implement Ford-Fulkerson (or Edmons-Karp) algorithm according to Algorithm 1. Implement a procedure for obtaining an initial feasible flow in networks with non-zero lower bounds using the transformation described in the lectures. Use this flow as the initial flow for the Ford-Fulkerson (Edmons-Karp) algorithm. Solve Survey Design problem as the Maximum Flow Problem by your algorithm on appropriately defined network. Upload source codes to CourseWare BRUTE System.

Hint 1: Formulate the Survey Design problem as Maximum Network Flow problem with non-zero lower bounds.

2.1 Input and Output Format

Your program will be called with two arguments: the first one is absolute path to input file and the second one is the absolute path to output file (the output file has to be created by your program).

The input file has the following format. On the first line, there are two natural numbers separated by space denoting the number of customers $|\mathcal{C}|$ and products $|\mathcal{P}|$. This is followed by $|\mathcal{C}|$ lines describing each customer. The customer i is described by two non-negative integers (might be zero) l_i and u_i, $l_i < u_i$ separated by spaces and followed by $|P_i|$ numbers $\{1, \ldots, |\mathcal{P}|\}$ describing which products the customer i can review. The input is concluded by line with $|\mathcal{P}|$ natural numbers giving review demands for each product $v_1, v_2, \ldots, v_{|\mathcal{P}|}$.

The output file consists of $|\mathcal{C}|$ lines (one line for each customer). Each line contains a list of numbers $\{1, \ldots, |\mathcal{P}|\}$ describing which product surveys were assigned to the customer i. If the input instance is infeasible (such a survey cannot be done), then write only -1.

Example 1

Input:

```
4 5
1 5 1 2 3
3 4 1 2 3 4 5
0 2 2 5
3 5 1 3 4 5
1 2 1 1 2
```

Output:

```
1 2 3
1 2 4 5
2 5
1 3 4 5
```

Example 2

Input:

```
2 2
0 1 1 2
0 1 1 2
3 3
```

Output:

```
-1
```