Data Introduction for Fixed Charge Network Flow Problems

Lu Yang
School of Data Science, University of Science and Technology of China

November 27, 2022

1 FCNF problems

Let $G = (V, E)$ be a directed graph with n nodes and m edges, where V is the set of nodes and E is the set of edges. Each edge $e \in E$ is associated with a flow x_e, a capacity u_e, a unit cost c_e, and a fixed cost s_e. The cost function $C_e(x_e)$ of edge e is discontinuous and can be expressed as:

$$C_e(x_e) = \begin{cases} 0 & x_e = 0 \\ s_e + c_e x_e & x_e \in (0, u_e] \end{cases} \tag{1}$$

Let $\mathbf{x}$ be the flow vector (decision variable), $\mathbf{A}$ be the node-edge incidence matrix of G, and $\mathbf{b}$ be the node supply vector. The fixed-charge network flow (FCNF) problem can be formulated as the following problem:

$$\begin{aligned} \min_{\mathbf{x}} \quad & C(\mathbf{x}) = \sum_{e \in E} C_e(x_e) \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & 0 \le \mathbf{x} \le \mathbf{u} \end{aligned} \tag{2}$$

2 Data Information

The data consist of two parts: randomly generating samples with various numbers of nodes and edges and a real-world scenario sample under decryption.

The randomly generating samples are divided into ten groups by the size of graph (# of nodes, # of edges). They are (20,100), (60,400), (120,1500), (150,2500), (200,4000), (250,7500), (300,9000), (500,10000), (1000, 20000), and (5000,100000). In each group, we construct 30 instances with the same size. In order to distinguish the different ratios of unit and fixed costs impacting on the model or solution, we multiply the fixed cost by an amplification factor in the first ten instances and multiply the unit cost by a reduction factor in the last ten instances. There factors are both setting as 10.

The real-world sample has been transformed into the same form of the randomly generating sample. The size of graph is (9909,26222).

3 Data Introduction

There are two datatype for the data: ".npy" and ".txt". The "npy" files can be loaded by the package "numpy" in Python, while the "txt" files can be loaded conveniently in C++ and Python.

For each instance, the data consist of two files: "A_bar_x.npy(txt)" and "uuf_x.npy(txt)", where the "x" in filename represents the item of this instance. In file "A_bar_x.npy(txt)", there is a matrix with size $(n, m+1)$, which is splicing by matrix $\mathbf{A}$ and vector $\mathbf{b}$ in Problem 2. In file "uuf_x.npy(txt)", there is a matrix with size $(3, m)$, which is splicing by vectors $\mathbf{u}$, $\mathbf{c}$, and $\mathbf{s}$.

Because of space limitation, I only upload txt files here. If you need npy files, please contact me by email: yl0501@mail.ustc.edu.cn