Discrete Applied Mathematics ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Inverse minimum flow problem under the weighted sum-type Hamming distance

Longcheng Liu *, Wenhao Zheng, Chao Li

School of Mathematical Sciences, Xiamen University, Xiamen 361005, PR China

ARTICLE INFO

Article history:
Received 13 April 2016
Received in revised form 6 January 2017
Accepted 5 June 2017
Available online xxxx

Keywords:
Minimum flow
Minimum cut
Inverse problem
Hamming distance
Strongly polynomial algorithm

ABSTRACT

The idea of the inverse optimization problem is to adjust the value of the parameters such that the observed feasible solution becomes optimal. The modification cost can be measured by different norms, such as l_1, l_2, l_∞ norms and the Hamming distance, and the goal is to adjust the parameters as little as possible.

In this paper, we consider the inverse minimum flow problem under the weighted sum-type Hamming distance, where the lower and upper bounds for the arcs should be changed as little as possible under the weighted sum-type Hamming distance such that a given feasible flow becomes a minimum flow. Two models are considered: the unbounded case and the general bounded case. We present their respective combinatorial algorithms that both run in $O(nm)$ time in terms of the minimum cut method. Computational examples are presented to illustrate our algorithms.

1. Introduction

Let $N(V, A, l, u, s, t)$ be a connected and directed network, where V $(|V| = n)$ is the node set, A $(|A| = m)$ is the arc set, l is the lower bound for the flow vector, u is the upper bound for the flow vector, s is the source node and t is the sink node. A *feasible* (s, t)*-flow* or simply *feasible flow* is a function $f : A \to R^{|A|}$ that satisfies the given lower and upper bounds

$$l_{ij} \le f_{ij} \le u_{ij} \quad \text{for each } (i, j) \in A,$$

and the flow conservation constraints

$$\sum_j f_{ij} - \sum_j f_{ji} = \begin{cases} v(f), & i = s, \\ -v(f), & i = t, \\ 0, & \text{other } i, \end{cases}$$

where $v(f)$ is the value of flow f from s to t. The minimum flow problem is to find a flow with the minimum flow value. It is well-known that the minimum flow problem has many applications in real world and can be solved in strongly polynomial time [1].

Let X and $\overline{X} = V \setminus X$ be a partition of all vertices such that $s \in X$ and $t \in \overline{X}$. An $s-t$ cut, denoted by $\{X, \overline{X}\}$, is the set of arcs with one endpoint in X and another endpoint in $\overline{X}$. We further use $(X, \overline{X})$ to express the set of forward arcs from a vertex in X to a vertex in $\overline{X}$ and use $(\overline{X}, X)$ to express the set of backward arcs from a vertex in $\overline{X}$ to a vertex in X. Then, the

* Corresponding author.
E-mail address: longchengliu@xmu.edu.cn (L. Liu).

http://dx.doi.org/10.1016/j.dam.2017.06.002

floor of an $s-t$ cut $\{X, \overline{X}\}$ is defined as follows:

$$\sum_{(i,j)\in(X,\overline{X})} l_{ij} - \sum_{(i,j)\in(\overline{X},X)} u_{ij}.$$

And the capacity of an $s-t$ cut $\{X, \overline{X}\}$ is defined as follows:

$$\sum_{(i,j)\in(X,\overline{X})} u_{ij} - \sum_{(i,j)\in(\overline{X},X)} l_{ij}.$$

If the network has the upper bound only, i.e., $l_{ij}=0$ for all $(i,j)\in A$, then the capacity of an $s-t$ cut $\{X, \overline{X}\}$ is simplified as

$$\sum_{(i,j)\in(X,\overline{X})} u_{ij}.$$

The following result is well known.

Lemma 1.1 (*[1]*). *A feasible flow f is a minimum flow if and only if its value equals the floor of an $s-t$ cut $\{X, \overline{X}\}$. In detail, we have*

$$f_{ij} = l_{ij}, \quad \text{if} \quad (i,j)\in(X,\overline{X}),$$

$$f_{ji} = u_{ji}, \quad \text{if} \quad (j,i)\in(\overline{X},X).$$

Conversely, an inverse minimum flow problem is to modify the lower bound or the upper bound of the flow vector as little as possible such that a given feasible flow becomes a minimum flow. Ciurua and Deaconu [4] considered the inverse minimum flow problem under the l_1 norm. They first presented a linear time and space method to decide whether the problem has solution, and then they presented a strongly polynomial algorithm with a time complexity $O(nm\log(n^2/m))$ and a weakly polynomial algorithm with a time complexity $O(\min\{n^{2/3}, m^{1/2}\}\cdot m\cdot\log(n^2/m)\cdot\log(\max\{n, R\}))$. At last, they considered some particular cases and presented a numerical example. Deaconu [5] considered the inverse minimum flow problem under the l_∞ norm, the inverse maximum flow problem with the upper bound only under the l_∞ norm and the inverse maximum flow problem with lower and upper bounds under the $O(m\log n)$ are presented. Furthermore, Deaconu [6] considered the inverse maximum flow problem with lower and upper bounds under the l_1 norm. A strongly polynomial algorithm with a time complexity $O(nm\log(n^2/m))$ and a weakly polynomial algorithm with a time complexity $O(\min\{n^{2/3}, m^{1/2}\}\cdot m\cdot \log(n^2/m)\cdot\log(\max\{n, R\}))$ are presented. Liu and Zhang [17] discussed inverse maximum flow problems under the weighted Hamming distance, for both sum-type and bottleneck-type, they presented strongly polynomial algorithms. Yang et al. [23] considered the inverse maximum flow problem under the l_1 norm and presented a strongly polynomial algorithm to solve it. Ahuja and Orlin [2] considered the inverse minimum cost flow problem under the l_1 and l_∞ norm. They transformed the unit weight inverse minimum cost flow problem under the l_1 norm into a unit capacity minimum cost circulation problem and transformed the unit weight inverse minimum cost flow problem under the l_∞ norm into a minimum mean cycle problem. For the general weight inverse minimum cost flow problem under the l_∞ norm, they transformed it into a minimum cost-to-weight ratio problem. Guler and Hamacher [9] considered the capacity inverse minimum cost flow problem. They showed it is NP-hard under the l_1 norm and presented an algorithm with a time complexity $O(nm^2)$ for the l_∞ norm. Jiang et al. [12] considered the inverse minimum cost flow problem under the weighted Hamming distance. They showed that the problem is APX-hard in the sum-type case and presented an algorithm with a time complexity $O(nm^2)$ for the bottleneck-type case. Furthermore, Tayyebi and Aman [20] presented a new algorithm with a time complexity $O(nm^2)$ for the inverse minimum cost flow problem under the bottleneck-type Hamming distance. Liu and Yao [16] considered the capacity inverse minimum cost flow problem under the weighted Hamming distance. They showed that the problem is APX-hard in the sum-type case and presented an algorithm with a time complexity $O(nm^2)$ for the bottleneck-type case. In this paper, we consider the inverse minimum flow problem under the weighted sum-type Hamming distance. Two variations will be discussed.

Let each arc (i,j) have an associated lower bound modification cost $w^l_{ij} > 0$ and an associated upper bound modification cost $w^u_{ij} > 0$. Let f^0 be a given feasible flow in the network $N(V, A, l, u, s, t)$. Then for the unbounded inverse minimum flow problem under the weighted sum-type Hamming distance, we look for a new lower bound l^* and a new upper bound u^* such that

(a) f^0 is a minimum flow of the network $N(V, A, l^*, u^*, s, t)$;

(b) the total modification cost of the arcs which are modified, i.e.,

$$\sum_{(i,j)\in A}\{w^l_{ij}H(l_{ij}, l^*_{ij}) + w^u_{ij}H(u_{ij}, u^*_{ij})\},$$

is minimized, where $H(x,y)$ is the Hamming distance between x and y, i.e., $H(x,y)=0$ if $x=y$ and 1 otherwise. Here, the weighted sum-type Hamming distance means that we use the sum of all the weighted Hamming distance between l_{ij} and l^*_{ij}, u_{ij} and u^*_{ij} to measure the modification.