European Journal of Operational Research 234 (2014) 916–920

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Short Communication

Note on "Inverse minimum cost flow problems under the weighted Hamming distance"

Javad Tayyebi *, Massoud Aman

Department of Mathematics, Faculty of Science, University of Birjand, Birjand, Iran

ARTICLE INFO

Article history:
Received 20 July 2013
Accepted 13 November 2013
Available online 25 November 2013

Keywords:
Combinatorial optimization
Minimum cost flow problem
Inverse problem
Hamming distance

ABSTRACT

Jiang et al. proposed an algorithm to solve the inverse minimum cost flow problems under the bottleneck-type weighted Hamming distance [Y. Jiang, L. Liu, B. Wuc, E. Yao, Inverse minimum cost flow problems under the weighted Hamming distance, European Journal of Operational Research 207 (2010) 50–54]. In this note, it is shown that their proposed algorithm does not solve correctly the inverse problem in the general case due to some incorrect results in that article. Then, a new algorithm is proposed to solve the inverse problem in strongly polynomial time. The algorithm uses the linear search technique and solves a shortest path problem in each iteration.

1. Introduction

The network $N(V, A, \mathbf{u}, \mathbf{c})$ is supposed where $V = \{1, 2, \dots, n\}$ is the set of nodes, A is the set of m arcs, $\mathbf{u}$ is the capacity vector for arcs and $\mathbf{c}$ is the cost vector for arcs. Each node $i \in V$ has an associated supply or demand of value $b(i)$. The well-known minimum cost flow (MCF) problem is formulated as follows (Ahuja, Magnanti, & Orlin, 1993):

$$\begin{aligned} &\min \sum_{(i,j)\in A} c_{ij}x_{ij}, \\ &\sum_{(i,j)\in A(i)} x_{ij} - \sum_{(j,i)\in A'(i)} x_{ji} = b(i) \quad \forall i \in N, \\ &0 \leqslant x_{ij} \leqslant u_{ij} \quad \forall (i,j) \in A, \end{aligned} \tag{1}$$

where $A(i)$ and $A'(i)$ are respectively the sets of arcs emanating from and arriving at node i. Assume that $\mathbf{x}^0$ is a feasible flow to the problem (1). The corresponding inverse problem is to modify components of the cost vector $\mathbf{c}$ as little as possible so that $\mathbf{x}^0$ becomes optimal to the problem (1). The modifications can be measured by various distances. Jiang, Liu, Wuc, and Yao (2010) considered the inverse minimum cost flow (IMCF) problem under the sum-type and the bottleneck-type weighted Hamming distance. In the sum-type case, they show that a special case of the inverse problem reduces to the weighted feedback arc set problem to be APX-hard. In the bottleneck-type case, they presented an algorithm to solve the inverse problem. In this note, we give a counter example to show that their algorithm fails for solving the IMCF problem in more cases. We show that this problem arises due to some incorrect results presented in that article. It is also mentioned that a restricted version of the IMCF problem can be solved correctly by this algorithm. Finally, we propose an algorithm based on the reduced cost optimality conditions to solve the problem in the general case. Our proposed algorithm solves a shortest path problem on an auxiliary network in each iteration and runs in strongly polynomial time.

2. Problem definition

In this section, the formulation of the IMCF problem is given and the algorithm proposed by Jiang et al. (2010) is reviewed.

For a feasible flow $\mathbf{x^0}$ of the MCF problem, its residual network $N'(V, A', \mathbf{u}', \mathbf{c}')$ can be constructed by the following algorithm.

Algorithm 1 (Ahuja et al., 1993).

Step 1: The node set is still V.
Step 2: If $(i,j) \in A$ and $x^0_{ij} < u_{ij}$, then $(i,j) \in A', c'_{ij} = c_{ij}$ and $u'_{ij} = u_{ij} - x^0_{ij}$.
Step 3: If $(i,j) \in A$ and $x^0_{ij} > 0$ then $(j,i) \in A', c'_{ji} = -c_{ij}$ and $u'_{ji} = x^0_{ij}$.

We denote the arc sets created by steps 2 and 3 as $A(c)_1$ and $A(c)_2$, respectively.

For each arc $(i,j) \in A$, the associated reduced cost is defined as $c^\pi_{ij} = c_{ij} - \pi_i + \pi_j$ where $\pi_i, i \in V$, is the ith variable of the corre-

* Corresponding author. Tel.: +98 0561 2502049; fax: +98 0561 4440340.
E-mail addresses: javadtayyebi@birjand.ac.ir, javadtayyebi@alum.sharif.ir (J. Tayyebi), mamann@birjand.ac.ir (M. Aman).

0377-2217/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2013.11.014

sponding dual problem. The following lemmas give some optimality conditions for a given feasible solution.

Lemma 2.1 (Negative cycle optimality conditions Ahuja et al., 1993). *A feasible flow* $\mathbf{x^0}$ *to the problem* (1) *is optimal if and only if the corresponding residual network* $N'(V, A', \mathbf{u}', \mathbf{c}')$ *does not contain any negative cost cycle.*

Lemma 2.2 (Reduced cost optimality conditions Ahuja et al., 1993). *A feasible flow* $\mathbf{x^0}$ *is optimal to the problem* (1) *if and only if some dual variables satisfy the conditions*

$$c^{\pi}_{ij} \geqslant 0, \quad \forall (i,j) \in A'.$$

Lemma 2.3 (Ahuja et al., 1993). *If* $\mathbf{x^0}$ *is optimal to the problem* (1), *then* $c^{\pi}_{ij} = 0$ *for each* $(i,j) \in A$ *with* $0 < x^0_{ij} < u_{ij}$.

From Lemma 2.1, the IMCF problem under the bottleneck-type weighted Hamming distance is formulated as follows:

$$\begin{aligned} &\min \max_{(i,j)\in A} w_{ij} H(d_{ij}, c_{ij}), \\ &\text{The residual network respect to } \mathbf{x^0} \text{ and} \\ &N(V, A, \mathbf{u}, \mathbf{d}) \text{ contains no negative cycle,} \\ &-p_{ij} \leqslant d_{ij} - c_{ij} \leqslant +q_{ij} \quad \forall (i,j) \in A, \end{aligned} \tag{2}$$

where for each $(i,j) \in A$, w_{ij} is a penalty associated with modifying c_{ij} to d_{ij}, p_{ij} and q_{ij} are respectively given bounds for decreasing and increasing cost coefficient c_{ij}, the value $H(d_{ij}, c_{ij})$ equals to 1 if $d_{ij} \neq c_{ij}$ and otherwise, $H(d_{ij}, c_{ij}) = 0$. Jiang et al. (2010) proposed the following algorithm to solve the problem (2).

Algorithm 2.

Step 1: Run Algorithm 1 to construct the residual network $N'(V, A', \mathbf{u}', \mathbf{c}')$ respect to $N(V, A, \mathbf{u}, \mathbf{c})$ and $\mathbf{x}^0$. Let $W = \Omega = \emptyset$; and go to Step 2.
Step 2: Choose a negative cost cycle C of the current residual network. If no negative cycle exists, then go to Step 5. Otherwise go to Step 3.
Step 3: If $C \setminus \Omega = \emptyset$, then go to Step 6. Otherwise go to Step 4.
Step 4: Find an arc $(x,y) \in C \setminus \Omega$ that satisfies

$$w_{xy} = \min\{w_{ij} : (i,j) \in C \setminus \Omega\} \tag{3}$$

and update the current network and the corresponding residual network as follows:
If $(x,y) \in A(c)_1$, then $c_{xy} = c_{xy} + q_{xy}, c'_{xy} = c_{xy}, W = W \cup \{w_{xy}\}$.
If $(x,y) \in A(c)_2$, then $c_{yx} = c_{yx} - p_{yx}, c'_{xy} = -c_{yx}, W = W \cup \{w_{yx}\}$. $\Omega = \Omega \cup \{(x,y)\}$.
Go back to Step 2.
Step 5: Stop and output that $\mathbf{x}^0$ is the minimum cost flow of the current network, the optimal solution of the problem (2) is the cost vector of the current network and the associated optimal objective value is

$$\max\{w_{ij} : w_{ij} \in W\}.$$

Step 6: Stop and output that the problem (2) has no feasible solution.

We shall mention that there exists a little typing error in Algorithm 2 (Jiang et al., 2010). In Step 4, they set $c'_{yx} = -c_{yx}$ when $(x,y) \in A(c)_2$. This can be corrected to $c'_{xy} = -c_{yx}$ since (x,y) is in the residual network and not (y,x).

3. A counter example

In this section, we give a counter example to show that Algorithm 2 does not solve the problem (2) in the general cases. Then, we state some reasons of the inaccuracy of the algorithm.

Example 3.1. The network given in Fig. 1(a) is considered. For each $(i,j) \in A$, the penalty w_{ij} is also given in Fig. 1(a). Assume that $p_{ij} = q_{ij} = 10$ for each $(i,j) \in A$. By using Algorithm 1, we construct the residual network respect to this network and feasible solution $x_{14} = 1$ and $x_{ij} = 0$ for all $(i,j) \in A \setminus \{(1,4)\}$ (see Fig. 1(b)). It is obvious that the residual network contains three negative cycle: 1–4–3–1, 1–2–4–1 and 1–2–4–3–1 are denoted by C_1, C_2 and C_3, respectively. It is easy to check an optimal solution of the problem (2) is $d^*_{14} = 5, d^*_{24} = c_{24} + q_{24} = 8$ and $d^*_{ij} = c_{ij}$ for every $(i,j) \in A \setminus \{(1,4),(2,4)\}$ with the objective value $w_{24} = \max\{w_{14}, w_{24}\} = 2$.

Now, let us to implement Algorithm 2 on this network. We initialize $\Omega = \emptyset, W = \emptyset$. Suppose that the algorithm identifies C_1 in the first iteration. Since $w_{14} = \min\{w_{ij} : (i,j) \in C_1 \setminus \Omega\}$ and $(1,4) \in A(c)_1$, the algorithm cancels this negative cycle by updating $c_{14} = c_{14} + q_{14} = 13$. Consequently, $c'_{14} = +13$ and $c'_{41} = -13$ in the new residual network. Also, $W = \{w_{14}\}$ and $\Omega = \{(1,4)\}$. It is easy

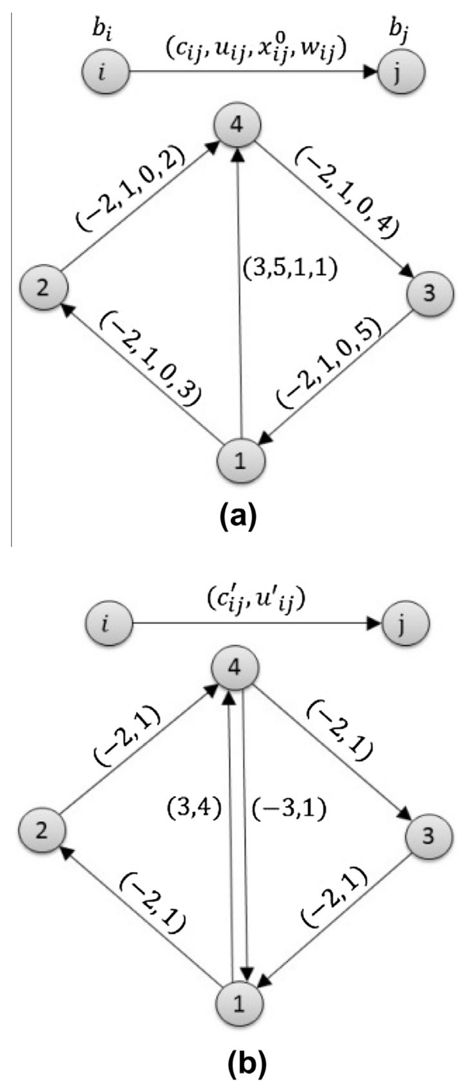

Fig. 1. (a) An instance of the minimum cost flow problem. (b) The corresponding residual network.