On inverse linear programming problems under the bottleneck-type weighted Hamming distance

Javad Tayyebi *, Massoud Aman

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

ARTICLE INFO

ABSTRACT

Given a linear programming problem with the objective function coefficients vector $\mathbf{c}$ and a feasible solution $\mathbf{x}^0$ to this problem, a corresponding inverse linear programming problem is to modify the vector $\mathbf{c}$ as little as possible to make $\mathbf{x}^0$ form an optimal solution to the linear programming problem. The modifications can be measured by different distances. In this article, we consider the inverse linear programming problem under the bottleneck-type weighted Hamming distance. We propose an algorithm based on the binary search technique to solve the problem. At each iteration, the algorithm has to solve a linear programming problem. We also consider the inverse minimum cost flow problem as a special case of the inverse linear programming problems and specialize the proposed method for solving this problem in strongly polynomial time. The specialized algorithm solves a shortest path problem at each iteration. It is shown that its complexity is better than the previous one.

1. Introduction

For an optimization problem and a given feasible solution $\mathbf{x}^0$ to this problem, an inverse problem is to adjust some parameters of the optimization problem as little as possible so that $\mathbf{x}^0$ becomes an optimal solution to the problem. The modifications can be measured by various distances as the l_1, l_2 and l_∞ norms and also, the weighted Hamming distances. The concept of inverse problems was first proposed by Tarantola [17] in geophysical sciences. In the context of optimization, Burton and Toint [5,6] considered the inverse shortest path problem using the l_2 norm which arises from applications in traffic modeling and seismic tomography. They formulated the problem as a convex quadratic programming problem.

A lot of inverse problems are studied when the modifications are measured by the (weighted) l_1, l_2 and l_∞ norms. Readers may refer to the survey paper [10] and papers cited therein. Ahuja and Orlin [2,3] considered the inverse linear programming problem under the l_1 and l_∞ norms and showed that the inverse problem of a linear programming problem is also a new linear programming problem and analyzed the inverse minimum cost flow problem as a special case. They showed that for the l_1 norm, the inverse minimum cost flow problem reduces to a unit capacity minimum cost flow problem and for the l_∞ norm, the problem is converted into a minimum cost-to-time ratio cycle problem. Zhang and Liu [19] considered some inverse linear programming problems under the l_1 norm, such as the inverse minimum cost flow problem and the inverse assignment problem, and presented strongly polynomial-time algorithms to solve the problems. Huang and Liu [11] proposed a method to solve the inverse linear programming problem under the l_p norm and then, they applied the method for solving the inverse minimum weight perfect k-matching problem defined on bipartite graphs.

---

* Corresponding author.
  E-mail addresses: javadtayyebi@birjand.ac.ir (J. Tayyebi), mamann@birjand.ac.ir (M. Aman).

All of the papers introduced in the preceding paragraph have used one of the l_1, l_2, l_∞ or l_p norms. However, it may be interesting to minimize the number of modified parameters, i.e., we might care about only whether a parameter is changed, but without considering the magnitude of its change. For this reason, the inverse optimization problems under the weighted Hamming distance also received attention. Unlike the norms, the Hamming distances are discontinuous and nonconvex. Therefore, the methods presented for the l_1, l_2 and l_∞ norms cannot be applied directly to solve the inverse problems under the Hamming distance.

Some inverse spanning tree problems under the weighted Hamming distances are studied in [9,13,14,20]. He et al. [9] showed that the inverse spanning tree problems under the sum-type Hamming distance reduces to the minimum node covering problem on an auxiliary bipartite network. Zhang et al. [20] achieved a similar result in the bottleneck-type case. In [13,14], the (constrained) inverse minmax spanning tree problems under the weighted Hamming distances has been studied and strongly polynomial-time algorithms have been presented. Liu and Zhang [16] considered the inverse maximum flow problem under the Hamming distances. They showed that the problem can be converted into a minimum cut problem and consequently, it can be solved in strongly polynomial time. Zhang et al. [21] considered two types of the inverse shortest path problems under the sum-type Hamming distance and showed both the problems are NP-hard. The inverse maximum perfect matching problems under the Hamming distances has been studied in [15]. In the sum-type case, the authors proved the problem is NP-hard. In the bottleneck-type, they proposed a strongly polynomial-time algorithm for solving the problem.

Two types of the inverse minimum cost flow problems are considered in the literature: the capacity inverse minimum cost flow problem and the (cost) inverse minimum cost flow problem. In the former, the capacity vector of a minimum cost flow problem is modified minimally so that a given feasible solution becomes optimal to the problem. This problem under the l_1 and l_∞ norms has been considered in [8]. For the l_1 norm, it is shown that the problem is NP-complete. For the l_∞ norm, a greedy algorithm is proposed. In the latter, the cost vector of a minimum cost flow problem is adjusted to make a given feasible solution form an optimal solution to the problem. Jiang et al. [12] studied this problem under the weighted Hamming distances (both the sum-type and bottleneck-type cases). In the sum-type case, they showed that the problem is APX-hard because a special case of the problem is equivalent to the weighted feedback arc set problem. In the bottleneck-type case, they proposed an $O(nm^2)$ algorithm for solving the problem. In [18], we showed that their proposed algorithm does not solve the inverse problem correctly. Then, we presented a new algorithm to solve the problem with the same complexity $O(nm^2)$.

To the best of your knowledge, despite considering too many types of inverse combinatorial problems under the Hamming distance in the literature, the generic inverse linear programming problem under the Hamming distance has not been considered yet. In this article, we consider this problem for the bottleneck-type case and propose an algorithm to solve it. The algorithm is based on the binary search technique. In [7], the authors have used the linear and binary search techniques to solve the inverse spanning tree problem, the inverse shortest path problem and the inverse assignment problem. Here, we extend their work to the inverse linear programming problem and the (cost) inverse minimum cost flow problem. For the inverse minimum cost flow problem, our proposed algorithm runs in $O(nm \log n)$ time which is better than the complexity of the algorithm presented by us in [18].

The rest of this paper is organized as follows. Section 2 gives some results on the inverse linear programming problem and an algorithm for solving the problem. Section 3 considers the inverse minimum cost flow problem as a special case of inverse linear programming problems. Finally, we conclude in Section 4.

2. Inverse linear programming problem

In this section, we formulate the inverse linear programming problem under the bottleneck-type weighted Hamming distance and we present an algorithm for solving the problem.

Consider a linear programming problem with m variables and n constraints as follows:

$$LP(\mathbf{c}) : \quad \min \quad z = \mathbf{c}^T \mathbf{x},$$
$$\text{s.t.} \quad A\mathbf{x} \le \mathbf{b},$$
$$\mathbf{x} \ge \mathbf{0},$$

where $\mathbf{c} \in \mathbb{R}^m$, $\mathbf{b} \in \mathbb{R}^n$ and $A = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m] \in \mathbb{R}^{n \times m}$ are given and $\mathbf{x} \in \mathbb{R}^m$ is to be determined. Suppose that $\mathbf{x}^0$ is a given basic feasible solution to the problem $LP(\mathbf{c})$ which is not optimal, necessarily. We want to modify the coefficients of the objective function as little as possible so that $\mathbf{x}^0$ becomes optimal to the modified problem. Assume that we incur an associated penalty w_i for modifying each coefficient c_i, $i = 1, 2, \ldots, m$. Therefore, in the inverse linear programming problem under the weighted bottleneck-type Hamming distance (ILPBH), we look for a vector $\hat{\mathbf{c}}$ so that the following conditions are satisfied:

- The solution $\mathbf{x}^0$ is optimal to the problem $LP(\hat{\mathbf{c}})$.
- $-l_i \le \hat{c}_i - c_i \le u_i$, $\forall i \in \{1, 2, , \ldots, m\}$, where $l_i \ge 0$ and $u_i \ge 0$ are respectively the given bounds for decreasing and increasing c_i. These constraints are called bound constraints.
- The value of $\max_{i=1,2,\ldots,m}\{w_i H(c_i, \hat{c}_i)\}$ is minimized, where $H(c_i, \hat{c}_i) = 0$ if $c_i = \hat{c}_i$ and $H(c_i, \hat{c}_i) = 1$ otherwise.