Journal of Global Optimization **15:** 213–218, 1999.

The Complexity Analysis of the Inverse Center Location Problem

M.C. CAI[1], X.G. YANG[1] and J.Z. ZHANG[2]
[1]*Institute of System Sciences, Academia Sinica, Beijing, China;* [2]*Department of Mathematics, City University of Hong Kong, Hong Kong*
(e-mail: mazhang@cityu.edu.hk)

(Received 18 May 1998; accepted in revised form 22 February 1999)

Abstract. Given a feasible solution, the inverse optimization problem is to modify some parameters of the original problem as little as possible, and sometimes also with bound restrictions on these adjustments, to make the feasible solution become an optimal solution under the new parameter values. So far it is unknown that for a problem which is solvable in polynomial time, whether its inverse problem is also solvable in polynomial time. In this note we answer this question by considering the inverse center location problem and show that even though the original problem is polynomially solvable, its inverse problem is NP–hard.

Key words: Complexity, Location problem, Networks and graphs, Satisfiability problem

1. Introduction

Recently there are quite a few papers discussing inverse network flow problems and inverse combinatorial optimization problems, see, for example, [1–5]. Given a feasible solution for a particular optimization problem, the inverse optimization problem is to modify some parameter values of the original problem as little as possible (under a certain norm), and sometimes even with other restrictions such as the bound constraints on the adjustment of some parameters, to make the given feasible solution become an optimal solution under the new parameter values. As shown in [3–5], the inverse problems of many combinatorial/network optimization problems can be solved by strongly or weakly polynomial algorithms. In fact in [6] it is shown that for a large class of combinatorial/network optimization problems, if the original problem can be solved in polynomial times, then its inverse problem can be solved in polynomial time by a quite uniform methodology. However, a very interesting problem still remains unsolved: is it true that the inverse problem is solvable in polynomial time whenever the original problem is solvable in polynomial time? In this note, we will claim that the answer to this question is **No!** We shall present an example, the inverse center location problem, which is NP-hard although its original problem has a strongly polynomial method to solve. With this example, we see that some inverse optimization problems are more difficult to solve than their original problems.

2. Inverse center location problem

The center location problem, which is to find the "best" location of a facility in a network to minimize the distance from the facility to the most remote vertex of the network, is a very practical problem and especially relevant to the problem of locating optimally a hospital, a police station, a fire station, or any other service facility. The center location problem is polynomially solvable [7].

We now consider the inverse center location problem, that is, to modify the weights of a network as little as possible under the l_1 norm such that a given vertex becomes a center of the network. We describe the inverse center location problem formally as follows.

Let $G = (V, A, w)$ be a directed and connected graph, where V is a vertex set, A is an arc set, $w : A \to R_+$ is a distance (weight) function. Let s be a specific vertex in V. The inverse center location problem is to change w to $w^* \geqslant 0$ such that

(a) s becomes a center of G under w^*, and

(b) the total adjustment $\sum_{e \in A} |w(e) - w^*(e)|$ is minimum.

The inverse center location problem has potential applications as it can be related to the regional development plan and show how to spend as less cost as possible to make an existing facility be 'relocated' to the center position. In this section, we try to show that the inverse center location problem is NP-hard. The technique which we use to reach this purpose is a polynomial time transformation of the SATISFIABILITY problem into an instance of a decision problem of the inverse center location problem.

A *Boolean variable* x is a variable that can assume only the values *true* and *false*. We call $\bar{x}$ the negation of x, where $\bar{x} = false$ if $x = true$, and $\bar{x} = true$ if $x = false$. A Boolean variable and its negation are both called *literals*. Given a group of Boolean variables, a *clause* consists of some literals of those Boolean variables. A clause is said *true* if one of its literals is *true*. The SATISFIABILITY problem [8] is as follows:

Given m clauses $\{C_1, C_2, \cdots, C_m\}$ involving n Boolean variables $\{x_1, x_2, \cdots, x_n\}$, is there a set of values for these Boolean variables (called a *truth assignment*) such that all clauses are *true*?

If such a set of values exists, we say the SATISFIABILITY problem is *satisfiable*.

The SATISFIABILITY problem is the earliest natural NP–complete problem proven by Cook [9].

THEOREM . *The inverse center location problem is NP-hard.*

Proof. Given an instance of the inverse center location problem $G = (V, A, w)$ and a number L, the decision problem of the inverse center location problem is whether there is a solution $w^* \geqslant 0$ (called a feasible solution) satisfying the request (a) above and

(c) $\sum_{e\in A} |w(e) - w^*(e)| \leqslant L$.

First, it is trivial to see that the decision problem of the inverse center location problem is in the NP class.

Let us now construct a reduction from the SATISFIABILITY problem to a decision problem of an inverse center location problem.

Consider an arbitrary instance of the SATISFIABILITY problem with n variables $\{x_1, x_2, \cdots, x_n\}$ and m clauses $\{C_1, C_2, \cdots, C_m\}$. Without loss of generality, we assume that, for each variable x_i and each clause C_j, x_i does not appear in C_j more than once, and x_i and $\bar{x}_i$ do not both appear in C_j.

Construct a digraph $G = (V, A)$ as follows:

$$V = \bigcup_{i=1}^{n}\{t_i, u_i, v_i\} \cup \{t_{n+1}, t_{n+2}, t_{n+3}\} \cup \bigcup_{j=1}^{m}\{q_j\},$$

$$\begin{aligned} A = &\bigcup_{i=1}^{n}\{(t_i, u_i), (t_i, v_i), (u_i, t_{i+1}), (v_i, t_{i+1})\} \cup \{(t_{n+1}, t_{n+2}), (t_{n+1}, t_{n+3})\} \\ &\cup \bigcup_{i=1}^{n}\bigcup_{j=1}^{m}(\{(u_i, q_j) | x_i \in C_j\} \cup \{(v_i, q_j) | \bar{x}_i \in C_j\}) \\ &\cup \bigcup_{i=1}^{n+1}\{(t_{n+2}, t_i), (t_{n+3}, t_i)\} \cup \bigcup_{i=1}^{n}\{(t_{n+2}, u_i), (t_{n+2}, v_i), (t_{n+3}, u_i), (t_{n+3}, v_i)\} \\ &\cup \bigcup_{j=1}^{m}\{(t_{n+2}, q_j), (t_{n+3}, q_j), (q_j, t_1)\} \cup \{(t_{n+2}, t_{n+3}), (t_{n+3}, t_{n+2})\}. \end{aligned}$$

In this digraph, the Boolean variable x_i corresponds to vertex u_i, its negation $\bar{x}_i$ corresponds to vertex v_i, and the clause C_j corresponds to vertex q_j. Further, there is an arc (u_i, q_j) if and only if $x_i \in C_j$, and (v_i, q_j) exists if and only if $\bar{x}_i \in C_j$. We call u_i and v_i literal vertices, q_j clause vertices, and the arcs (u_i, q_j) or (v_i, q_j) clause arcs. Note that from t_{n+2} and t_{n+3}, there are arcs to all other vertices. Let $s = t_1$, define a weight function on G as

$$w(e) = \begin{cases} 0 & e = (u_i, t_{i+1}) \text{ or } (v_i, t_{i+1}), \\ 1 & \text{otherwise.} \end{cases}$$

Note that the distances from t_{n+2} (and t_{n+3}) to all other vertices are 1.

Now we claim that the SATISFIABILITY problem is satisfiable if and only if there is a feasible solution for the decision problem of the above inverse center location problem whose total absolute adjustment of the weights is at most n (i.e. $L = n$ in (c)).

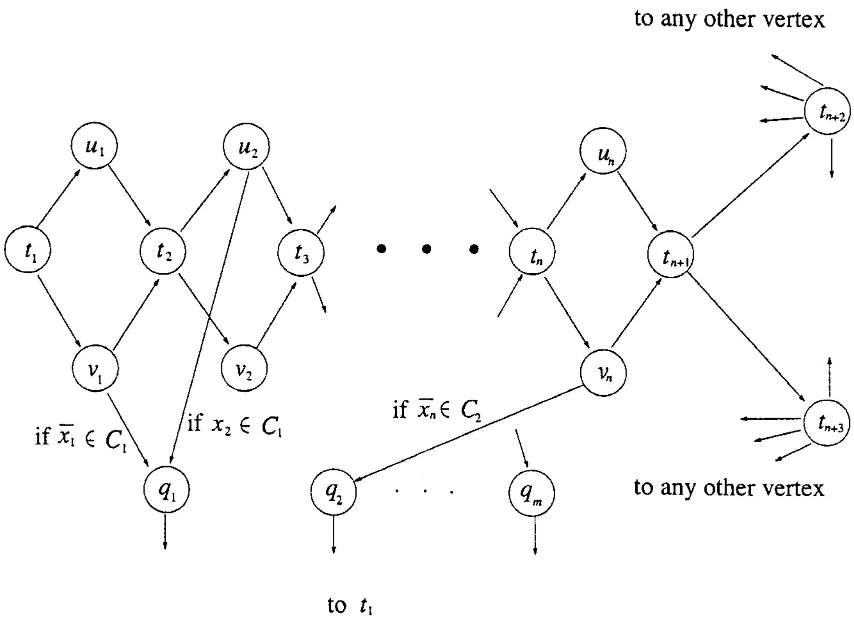

Figure 1. The digraph G

We first assume that the SATISFIABILITY problem is satisfiable. Let $P^* = t_1 z_1 t_2 z_2 \ldots t_i z_i \ldots t_{n+1}$ denote the path from t_1 to t_{n+1} where z_i is the vertex corresponding to the true literal, that is, z_i is either vertex u_i or v_i according to $x_i = \textit{true}$ or *false*, $i = 1, 2, \ldots, n$. For example, if $x_1 = \textit{false}$ and $x_2 =$ true, then the part of the path from t_1 to t_3 would be $t_1 v_1 t_2 u_2 t_3$. Let us change the weights of the arcs on path P^* to zero to obtain the new weight vector w^*. Obviously, $\sum_{e \in A} |w(e) - w^*(e)| = n$. We denote by $d_{w^*}(t_1, p)$ the distance from t_1 to a vertex $p \in V(G)$ under w^*. Then $d_{w^*}(t_1, p) \leqslant 1$ for every vertex $p \in V(G)$. To justify this, first it is clear that $d_{w^*}(t_1, p) = 0$ for each vertex p on P^*. Hence $d_{w^*}(t_1, t_{n+2}) \leqslant 1$, $d_{w^*}(t_1, t_{n+3}) \leqslant 1$, and $d_{w^*}(t_1, z_i') \leqslant 1$ for each vertex z_i' (since $d_{w^*}(t_1, t_i) = 0$) where z_i' denotes either vertex v_i or u_i not on P^*, $i = 1, 2, \ldots, n$. And for each clause vertex q_j, as there exists at least one true literal, say ℓ_i, in clause C_j, and the vertex corresponding to ℓ_i is on P^*, implying $d_{w^*}(t_1, q_j) \leqslant 1$. Hence the claim is true. But from any vertex other than t_1, its largest distance to other vertices is at least 1. So, vertex t_1 becomes a center of G under the new weight w^*. In other words, w^* is a feasible solution satisfying (a) and (c) with $L = n$.

Conversely, suppose that there exists a solution w^* such that t_1 becomes a center of G under w^* and $\sum_{e \in A} |w(e) - w^*(e)| \leqslant n$. We need to find a truth assignment to make all clauses true.

Denote by $d_w(u, v)$ the distance from vertex u to vertex v under w, which is equal to the length of the shortest path from u to v under w, and denote by $d_w(u)$ the largest distance from u to other vertices under w.

Let $r = d_{w^*}(t_1)$. Since $d_w(t_1, t_{n+2}) = n + 1$, and the total adjustment of the weights cannot be more than n, of course $r \geqslant 1$. On the other hand, as $d_w(t_{n+2}) = 1$, we know that $d_{w^*}(t_{n+2}) \leqslant n+1$. Since t_1 is now a center, $r \leqslant d_{w^*}(t_{n+2}) \leqslant n+1$. As all acyclic paths from t_1 to t_{n+2} have length $n + 1$ under w, but $d_{w^*}(t_1, t_{n+2}) \leqslant d_{w^*}(t_1) = r$, we need to shorten at least one path P from t_1 to t_{n+2} by length $x = n + 1 - r$ or more. The fact that under w all arcs from t_{n+2} has unit length means that we need to extend one arc from t_{n+2} by at least length $r - 1 = n - x$ to ensure $d_{w^*}(t_{n+2}) \geqslant r$. Similarly, we should also extend one arc from t_{n+3} by at least length $n - x$ to ensure $d_{w^*}(t_{n+3}) \geqslant r$. So, the total modification of the weights is at least $x + 2(n - x) = 2n - x$. We claim that $x = n$. In fact, from $r \geqslant 1$, we have $x \leqslant n$. On the other hand, from $2n - x \leqslant \sum_{e \in E} |w(e) - w^*(e)| \leqslant n$, we have $x \geqslant n$. Thus $x = n$, from which we can easily deduce that $r = 1$, and only one acyclic path from t_1 to t_{n+2} can be modified.

Furthermore, we know that no adjustment can be made on (t_{n+1}, t_{n+2}) and (t_{n+1}, t_{n+3}). In fact if say, the arc (t_{n+1}, t_{n+2}) is shortened by $\delta > 0$, then the distance from t_1 to t_{n+3} under w^* is at least $(n + 1) - (n - \delta) = 1 + \delta$. So, $r \geqslant d_{w^*}(t_1, t_{n+3}) \geqslant 1+\delta$ which conflicts with $r = 1$. Therefore we can only modify one path from t_1 to t_{n+1}. Such a path consists of one and only one route from t_i to t_{i+1} in the pair $\{A_i^+, A_i^-\}$ for $i = 1, 2, \cdots, n$, where $A_i^+ = \{(t_i, u_i), (u_i, t_{i+1})\}$ and $A_i^- = \{(t_i, v_i), (v_i, t_{i+1})\}$. Note that if A_i^+ is on the path, the weight of (t_i, u_i) is reduced from 1 to 0; if the path contains A_i^-, the weight of (t_i, v_i) becomes 0; and these are the only changes of the original weights. This path corresponds to a truth assignment of the SATISFIABILITY problem. That is, $x_i = true$ if A_i^+ is on the path, and $x_i = false$ otherwise.

It is easy to show that such a truth assignment can guarantee that each clause C_j is *true*. In fact, for any clause vertex q_j, if all the literal vertices connecting to the clause vertex q_j are not on the above mentioned path from t_1 to t_{n+1}, then the length of any path from t_1 and q_j is at least 2. This contradicts the proven fact $r = 1$. Hence we know that at least one literal vertex connecting to the clause vertex q_j must be on the path, and this literal vertex just corresponds to a *true* literal of clause C_j, and hence clause C_j is *true*. In other words, we conclude that the SATISFIABILITY problem is satisfiable.

As the number of vertices is $m+3n+3$, and the number of arcs does not exceed $mn + 10n + 3m + 6$, the transformation from the SATISFIABILITY problem to the decision problem of the inverse center location problem is indeed a polynomial reduction. Thus the inverse center location problem is NP-hard.

The proof is completed. □

3. Conclusion

From this note, we see that the inverse problem of a polynomially solvable problem is not necessarily polynomially solvable and may become a more challenging NP-hard problem. Roughly speaking, only if the feasible region of the inverse problem is a polytope, can we ensure that the l_1 inverse problem is solvable in polynomial time. For example, the inverse linear programming problem is this kind. Unfortunately, for the inverse location problem discussed in this note, its feasible region cannot be characterised by a set of linear equations/inequalities. As to various types of inverse center location problems, we shall propose effective solution methods in forthcoming papers.

Acknowledgments

The authors gratefully acknowledge the partial support of the Hong Kong Universities Grant Coucil (CERG CITYU-9040189).

References

1. Burton, D. and Toint, Ph.L., (1992). On an instance of the inverse shortest paths problem, *Mathematical Programming*, 53: 45–61.
2. Burton, D., Pulleyblank, B. and Toint, Ph.L. (1997). The inverse shortest path problem with upper bounds on shortest path costs, in *Network Optimization*, edited by P.M. Pardalos, D.H. Hearn and W.H. Hager, *Lecture Notes in Economics and Mathematical Systems* 450: 156–171.
3. Zhang, J.Z. and Liu, Z.H. (1996). Calculating some inverse linear programming problems, *J. Computational and Applied Mathematics* 72: 261–273.
4. Zhang, J.Z. and Ma, Z.F. (1996). A network flow method for solving some inverse combinatorial optimization problems, *Optimization* 37: 59–72.
5. Cai, M.C. and Li, Y.J. (1997). Inverse matroid intersection problem, *ZOR Math. Meth. Oper. Res.* 45: 235–243.
6. Yang, C. and Zhang, J. Z. (1999). Two general methods for inverse optimization problems, *Appl. Math. Lett.* 12(2): 69–72.
7. Christofides, N. (1975). *Graph Theory: An Algorithmic Approach*, Academic Press, London.
8. Papadimitriou, C.H. and Steiglitz, K. (1982). *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, New Jersey.
9. Cook, S.A. (1971). *The complexity of theorem proving procedures*, Proceedings of the 3rd ACM Symposium on the Theory of Computing, 151–158.