Journal of Global Optimization (2006) 34: 551–567
DOI 10.1007/s10898-005-1649-y

A Class of Inverse Dominant Problems under Weighted l_∞ Norm and an Improved Complexity Bound for Radzik's Algorithm*

QIN WANG[1], XIAOGUANG YANG[2,†] and JIANZHONG ZHANG[3]
[1]*Department of Mathematics, China Jiliang University, Hangzhou 310018, P. R. China*
[2]*Institute of Systems Science, Chinese Academy of Sciences, Beijing 100080, P. R. China*
[3]*Department of Mathematics, City University of Hong Kong, Hong Kong, P. R. China*

(Received 27 May 2003; accepted 27 July 2005)

Abstract. In this paper, we first discuss a class of inverse dominant problems under weighted l_∞ norm, which is how to change the original weights of elements with bounds in a finite ground set so that a given set becomes a weakly dominant set with respect to a given collection of subsets under the new weights and the largest change of the weights is minimum. This model includes a large class of improvement problems in combinatorial optimization. We propose a Newton-type algorithm for the model. This algorithm can solve the model in strongly polynomial time if the subproblem involved is solvable in strongly polynomial time. In the second part of the paper, we improve the complexity bound for Radzik's Newton-type method which is designed to solve linear fractional combinatorial optimization problems. As Radzik's method is closely related to our algorithm, this bound also estimates the complexity of our algorithm.

Key words: dominant set, improvement problem, inverse problem, linear fractional combinatorial optimization, polynomial time algorithm

1. Introduction

Since Burton and Toint published their paper on an instance of the inverse shortest paths problem [2], there has been a number of papers concerning *inverse optimization problems*. Heuberger gave a comprehensive survey on the subject [5]. Typically, an inverse optimization problem is to modify the coefficients of the objective function such that a given solution becomes an optimal one under the modified coefficients and the deviation of the modified coefficients from the original ones is minimized. For example, the inverse minimum spanning tree problem is to change the lengths of edges in a network as little as possible so that a given spanning tree T' becomes

*Supported by the Hong Kong Universities Grant Council (CERG CITYU 9040883 and 9041091).

†The author is also grateful for the support by the National Key Research and Development Program of China (Grant No. 2002CB312004) and the National Natural Science Foundation of China (Grant No. 70425004).

the minimum length spanning tree under the revised lengths. The frequently used measurements for minimum deviation (or minimum cost) in handling such problems are l_1, l_∞ and l_2 norms of the deviation vector. Under weighted l_∞ norm, Zhang and Liu proposed a general model for inverse combinatorial optimization problems by introducing a concept "dominant set" (we refer it as "strongly dominant set" in this paper) [10]. They proved that many inverse optimization problems are special cases of this model, such as inverse spanning tree problem, inverse maximum weight matching problem, inverse minimum cost flow problem, etc.

Recently, a type of closely related problems also attracted much attention which requests us to change the coefficients of the objective function as little as possible under certain measurement, such that the optimal value under the modified coefficients is not greater (or less) than a given level. To make the terminology clear, we call such a problem an *improvement problem*. For example, a spanning tree improvement problem is to change the lengths of edges in a network as little as possible so that the total length of the minimum spanning tree under the new length vector is upper bounded by a given value c.

Under l_1 norm, Burton et al. [1], Fekete et al. [3], and Zhang and Lin [9] considered the shortest path improvement problem which is to modify the weights minimally such that distances between some given node pairs are within required upper bounds. They showed that this shortest path improvement problem is NP-hard, and therefore, only some special cases can be solved polynomially. Note that what they discussed is the multi-objective case, i.e., the lengths between several pairs all meet given bounds. If we consider only one objective, say the distance between one pair of nodes, or the total weight of a spanning tree, then the improvement problem would be relatively easier to solve. However, if bound restrictions on the modification of weights are imposed, then even for some simple optimization problems, such as the shortest path problem, the minimum spanning tree problem, the assignment problem and the minimum cut problem, their l_1-norm improvement problems are still NP-hard.

Under weighted l_∞ norm, it appears that some improvement problems cannot be handled by Zhang and Liu's model in [10]. In this paper, we propose a new dominant relationship: weakly dominant set. The new model includes a class of improvement problems under weighted l_∞ norm. We discuss the solvability of the new model under bound restrictions. We show that this new model can be solved in strongly polynomial time if the subproblem involved in the computation is solvable in strongly polynomial time for any fixed value of the parameter which appears in the subproblem.

The core algorithm for solving this model can be considered as a discrete type Newton method, which shares the same properties with Radzik's algorithm [8]. Radzik's algorithm is designed for solving linear fractional combinatorial optimization (LFCO) problems. Radzik gave a strongly polynomial bound, $O(p^2 \log^2 p)$, for his method, where p is the size of input. In the second part of our paper, we prove that the complexity bound of Radzik's algorithm can be improved to $O(p^2 \log p)$. This implies that Radzik's algorithm as well as our algorithm run faster than expected before.

In Section 2, we will describe the general model under weighted l_∞ norm and propose an algorithm for solving the model. Then in Section 3, we will discuss how to improve the complexity bound of Radzik's method. Throughout the paper, for a set F, we use $|F|$ to represent its cardinality.

2. A General Model for a Class of Inverse Dominant Problems and the Solution Procedure

Let $E = \{e_1, e_2, \dots, e_n\}$ be a finite ground set and each element $e_i \in E$ be associated with a non-negative weight $w(e_i)$. Let $\mathcal{F}$ be a family of subsets of E, $(S, \bar{S})$ be a given partition of E (the special cases that $S = E$ or $S = \emptyset$ are allowable). For each $F \in \mathcal{F}$ we define a *deficiency* of F with respect to w as

$$d_w(F) = \sum_{e \in F \cap \bar{S}} w(e) - \sum_{e \in F \cap S} w(e).$$

Let c be a given real value. In [10], S is called a *dominant set* with respect to $\mathcal{F}$ under w if $d_w(F) \leqslant c$ for all $F \in \mathcal{F}$. We may refer to so defined dominant set as a *strongly dominant set*. Zhang and Liu [10] considered an inverse model that is how to make a given set S become a strongly dominant set by adjusting the weights of e_i in the ground set. They showed that a large class of inverse combinatorial problems under l_∞ norm can be embedded into this model.

In this paper, we consider a new dominant relationship, weakly dominant set. It can be defined as follows.

DEFINITION 2.1. Let $(S, \bar{S})$ be a given partition of E. For each $F \in \mathcal{F}$ a deficiency $d_w(F)$ with respect to w is defined as above, and if there is an $F \in \mathcal{F}$ such that $d_w(F) \leqslant c$, then S is called a *weakly dominant set* with respect to $\mathcal{F}$ under w.

Similar to the problem discussed in [10], we define an inverse (weakly) dominant problem as follows: if the given S is not a weakly dominant set with respect to $\mathcal{F}$ under w, we want to change the weight vector w as little as possible such that S becomes a weakly dominant set under the changed weight vector $\bar{w}$.

The 'minimum change' in vector w is characterized in this paper by a weighted l_∞ norm, and we also impose bound restrictions on the change (increase or decrease) of w. To say more formally, let b: $E \to R_+$ be a bound function by which the changes of w are restricted (assume $b \leqslant w$), and let p: $E \to R_+$ be the cost vector for per unit of change in w, the inverse dominant problem under weighted l_∞ norm can be stated as follows.
(IP) Find an adjusted weight vector $\bar{w} \geqslant 0$ such that

(a) the solution $F_{\bar{w}}$ of the minimization problem OP($\bar{w}$) below meets the condition $d_{\bar{w}}(F_{\bar{w}}) \leqslant c$,
(b) $|w(e) - \overline{w}(e)| \leqslant b(e)$ for each $e \in E$, and
(c) the modification cost $\|w - \overline{w}\|_\infty^{(p)} := \max\{p(e)|w(e) - \overline{w}(e)| \mid e \in E\}$ is minimum, where the minimization problem is

(OP($\overline{w}$)) $\min\{d_{\bar{w}}(F) \mid F \in \mathcal{F}\}$.

We now present an example of the inverse dominant problem (IP). Suppose that in a telecommunication network each link is represented by an edge, and a company has already possessed some links in the network. Let us denote this set of links by S and call them old links. Now the company wants to improve its utility by selling some old links and buying some new links from $\bar{S}$. For each $e \in E = S \cup \bar{S}$, let $w(e)$ be the estimate price of the link. Let c be the available budget for the purpose. A prerequisite requirement of the company is to cover all nodes of the network, that is, the set of links which the company shall own after the trade should contain a spanning tree. This problem can be formulated as a weakly dominant relationship:

Let $\mathcal{T}$ be the set of all spanning trees. For each $T \in \mathcal{T}$, define the symmetric difference of T and S by $F(T) = (T \backslash S) \cup (S \backslash T)$. Let $\mathcal{F} = \{F(T) | T \in \mathcal{T}\}$. For each $F(T) \in \mathcal{F}$, consider the deficiency of $F(T)$,

$$\begin{aligned} d_w(F(T)) &= \sum_{e \in F(T) \cap \bar{S}} w(e) - \sum_{e \in F(T) \cap S} w(e) \\ &= \sum_{e \in T \cap \bar{S}} w(e) - \sum_{e \in S \backslash T} w(e). \end{aligned}$$

The first sum on the right-hand side is the total expenditure spent on buying new links, while the second sum is the total return by selling unnecessary old links, i.e., the old links out of T. Therefore $d_w(F(T))$ is the net cost under the constraint that the company should possess all links in T. The weakly dominant relationship in fact reflects that under its budget limitation, the company is able to have a spanning tree of links after a trade.

The weakly dominant model can be used for another purpose. Note that $-d_w(F(T)) = \sum_{e \in F(T) \cap S} w(e) - \sum_{e \in F(T) \cap \bar{S}} w(e)$ is the total gain from the selling–buying activity. Assume that the company has a profit goal, say p.

Let $c = -p < 0$. Then $d_w(F(T)) \leqslant c$ is equivalent to $-d_w(F(T)) \geqslant p$. In this case, the weakly dominant relationship means that the company can make profit p or more by selling some old links and purchasing some new links, meanwhile let the owned links still contain a spanning tree.

Now consider the inverse problem. Suppose that under the estimated prices $\{w(e) | e \in E\}$, S is not a weakly dominant set with respect to $\mathcal{F}$, i.e., the company cannot get a spanning tree of links under the budget constraint. So, the company tries to set some marketing strategy to realize its goal, that is, to decide a set of targeted buying/selling prices w' with which to negotiate with its business counterparts. Obviously, w' cannot be too big or too small, and should meet some bound constraints: $|w(e) - w'(e)| \leqslant b(e)$, where b is a given bound vector. Obviously, a realistic marketing strategy is that instead of changing prices of only a small part of edges but with large amounts, one should consider a balanced price change on some edges, even if more edges would be involved. This means that an ideal adjustment is to make the maximum change minimal. Therefore, a reasonable model should determine w' as a solution of the problem

$$\begin{aligned} &\min \max_{e \in E} |w(e) - w'(e)| \\ &\text{s.t. } \min\{d_{w'}(F) | F \in \mathcal{F}\} \leqslant c, \\ &\qquad |w(e) - w'(e)| \leqslant b(e), \quad e \in E, \end{aligned}$$

which is just a special case of problem (IP) when all $p(e)$ are equal. Note that the first constraint asks S to become a weakly dominant set under w'.

We note that:

(1) The model (IP) in this paper is different from the model in [10]. Let us take the minimum spanning tree problem as an example. If we let $\mathcal{F}$ be the set of all spanning trees and choose $S = \emptyset$, then this paper deals with the minimum spanning tree improvement problem which is how to modify the weights by minimum cost such that the length of the minimum spanning tree is within a given level c. While [10] considers how to make minimum change for the weight vector such that a given spanning tree (which is not the minimum one under the original weight) becomes a minimum spanning tree under the new weight. Obviously neither one can include the other model as a special case.

(2) Generally speaking, our problem in this paper is an inverse problem: to make a given set S which is not a dominant one become a dominant set, but on the other hand by the definition of weak dominance, this problem can also be regarded as an improvement problem: make a minimum cost change in weight such that we are able to find a subset $\bar{F}$ in $\mathcal{F}$ which makes the objective value under the new weight vector $\overline{w}$, i.e., $d_{\overline{w}}(\bar{F})$, be upper bounded by a given value c. As we can

define different $\mathcal{F}$ and S and set different values for c, the model of this paper can include various improvement problems.

(3) The algorithm for solving the model in [10] may yield negative weights. In this paper, as we impose bound restrictions, the model can avoid the negative weight case.

We now consider how to solve problem (IP). Let

$$\hat{w}(e)=\begin{cases} w(e)+b(e) & \text{for } e\in S,\\ w(e)-b(e) & \text{for } e\in \bar{S}.\end{cases}$$

Then the inverse dominant problem is feasible if and only if $d_{\hat{w}}(F_{\hat{w}})\leqslant c$. Moreover, if $d_w(F_w)\leqslant c$, it is clear that we need not change any weight. In the sequel, we always assume that $d_{\hat{w}}(F_{\hat{w}})\leqslant c$ and $d_w(F_w)>c$.

For any non-negative real number r, define a modified weight function as

$$w_r(e)=\begin{cases} \min\{w(e)+\frac{r}{p(e)},\hat{w}(e)\} & \text{for } e\in S,\\ \max\{w(e)-\frac{r}{p(e)},\hat{w}(e)\} & \text{for } e\in \bar{S}.\end{cases}$$

Clearly, the modification cost of $w_r(e)$ under weighted l_∞ norm is bounded by r: $\|w_r-w\|_\infty^{(p)}\leqslant r$. In other words, if the budget is r, then the best we can do is to adjust w to w_r. It is intuitive that the (IP) under weighted l_∞ norm is equivalent to finding a minimum r^* such that

$$d_{w_{r^*}}(F_{w_{r^*}})=c. \tag{1}$$

Suppose we have an algorithm $\mathcal{A}$ to solve the problem (OP(w')) for a fixed weight vector w' (problem (OP(w')) is defined by replacing $\overline{w}$ in problem (OP($\overline{w}$)) by w'). In this section, we will design a general solution procedure for solving problem (IP). The procedure can be divided into two phases.

Phase 1. Phase 1 consists of two steps. For each $e\in E$, write the weighted bound $\tilde{b}(e)=p(e)b(e)$. Let $\tilde{b}_0=0$. The first step in Phase 1 is to sort the elements in a strictly increasing order according to the weighted bounds $\{\tilde{b}(e)\}$, say $\tilde{b}_0<\tilde{b}_1<\tilde{b}_2<\cdots<\tilde{b}_m$ ($m\leqslant|E|$).

It is straightforward to see that $w_{\tilde{b}_m}(e)=\hat{w}(e)$ for each $e\in E$. By the assumption that $d_{\hat{w}}(F_{\hat{w}})\leqslant c$, i.e. $d_{w_{\tilde{b}_m}}(F_{w_{\tilde{b}_m}})\leqslant c$, we have $r^*\leqslant\tilde{b}_m$.

We claim that

THEOREM 2.1. *If* $d_{w_{\tilde{b}_{k-1}}}(F_{w_{\tilde{b}_{k-1}}}) > c$ *and* $d_{w_{\tilde{b}_k}}(F_{w_{\tilde{b}_k}}) = c$ *for some* $k(1 \leqslant k \leqslant m)$, *then* $r^* = \tilde{b}_k$.

Proof. It is straightforward to see that $\tilde{b}_{k-1} < r^* \leqslant \tilde{b}_k$ since $d_{w_{\tilde{b}_{k-1}}}(F_{w_{\tilde{b}_{k-1}}}) > c$ and $d_{w_{\tilde{b}_k}}(F_{w_{\tilde{b}_k}}) = c$.

Let $E^- = \{e \in E \mid \tilde{b}(e) \geqslant \tilde{b}_k\}$. We claim that $F_{w_{r^*}} \cap E^- \neq \emptyset$. For otherwise, for every $e \in F_{w_{r^*}}$ we would have $\tilde{b}(e) \leqslant \tilde{b}_{k-1} < r^*$, which implies that

$$w_{r^*}(e) = w_{\tilde{b}_{k-1}}(e) \quad \text{for each } e \in F_{w_{r^*}}.$$

Hence, we have

$$\begin{aligned} d_{w_{\tilde{b}_{k-1}}}(F_{w_{\tilde{b}_{k-1}}}) &\leqslant d_{w_{\tilde{b}_{k-1}}}(F_{w_{r^*}}) \\ &= d_{w_{r^*}}(F_{w_{r^*}}) = c. \end{aligned}$$

This is a contradiction with the assumption $d_{w_{\tilde{b}_{k-1}}}(F_{w_{\tilde{b}_{k-1}}}) > c$. So, $F_{w_{r^*}} \cap E^- \neq \emptyset$.

For any r' and r such that $\tilde{b}_{k-1} \leqslant r' < r \leqslant \tilde{b}_k$, it is easy to see that, for each $e \in E^-$, $w_{r'}(e) < w_r(e)$ for $e \in S$; and $w_{r'}(e) > w_r(e)$ for $e \in \bar{S}$. Therefore for any $F \in \mathcal{F}$ such that $F \cap E^- \neq \emptyset$, we have $d_{w_{r'}}(F) > d_{w_r}(F)$.

Now if $r^* < \tilde{b}_k$, we obtain that $c = d_{w_{r^*}}(F_{w_{r^*}}) > d_{w_{\tilde{b}_k}}(F_{w_{r^*}}) \geqslant d_{w_{\tilde{b}_k}}(F_{w_{\tilde{b}_k}})$, contradicting the assumption $d_{w_{\tilde{b}_k}}(F_{w_{\tilde{b}_k}}) = c$. Therefore, $r^* < \tilde{b}_k$ is impossible, and we must have $r^* = \tilde{b}_k$. □

The second step in Phase 1 is to use the binary search technique to find a k, $1 \leqslant k \leqslant m$, such that $d_{w_{\tilde{b}_{k-1}}}(F_{w_{\tilde{b}_{k-1}}}) > c$ and $d_{w_{\tilde{b}_k}}(F_{w_{\tilde{b}_k}}) \leqslant c$.

If $d_{w_{\tilde{b}_k}}(F_{w_{\tilde{b}_k}}) = c$, we have $r^* = \tilde{b}_k$ by Theorem 2.1, and hence the optimal solution has already been obtained. Otherwise we have $\tilde{b}_{k-1} < r^* < \tilde{b}_k$. In this case, we should go to the next phase.

Phase 2. The computation of Phase 2 can be described as the following algorithm.
Algorithm:

Step 0. Set $\bar{r} = \tilde{b}_k$.
Step 1. Let $\delta = \frac{c - d_{w_{\bar{r}}}(F_{w_{\bar{r}}})}{\mu(F_{w_{\bar{r}}} \cap E^-)}$, and update $r = \bar{r} - \delta$, where $\mu(e) = \frac{1}{p(e)}$ for each $e \in E$, and $\mu(X) = \sum_{e \in X} \mu(e)$ for any subset $X \subset E$.
Step 2. If $d_{w_r}(F_{w_r}) = c$, stop, and the optimal solution is $\bar{w}(e) = w_r(e)$ for all $e \in E$.
Step 3. Otherwise let $\bar{r} = r$, go to Step 1.

Let us check the validity of the algorithm.

First, we claim that the algorithm is well-defined. To this purpose we need to show that the following four conclusions are true at Step 1:

(a) $d_{w_{\bar{r}}}(F_{w_{\bar{r}}}) < c$,
(b) $F_{w_{\bar{r}}} \cap E^- \neq \emptyset$,
(c) $\tilde{b}_{k-1} < r < \bar{r} \leqslant \tilde{b}_k$,
(d) $d_{w_r}(F_{w_r}) \leqslant c$.

In fact for the first time to execute Step 1, which directly follows Step 0 with $\bar{r} = \tilde{b}_k$, $d_{w_{\bar{r}}}(F_{w_{\bar{r}}}) = d_{w_{\tilde{b}_k}}(F_{w_{\tilde{b}_k}}) < c$. Since $\bar{r} > \tilde{b}_{k-1}$, we can prove $F_{w_{\bar{r}}} \cap E^- \neq \emptyset$ using the method shown in Theorem 2.1, but replacing r^* there by $\bar{r}$. In other words, in the first iteration, (a) and (b) hold.

In what follows, we show that when (a) and (b) hold, (c) and (d) also hold in the same iteration; and when (c) and (d) are true, (a) and (b) in the next iteration are also true if the computation does not stop.

When (a) and (b) hold, $\delta > 0$ and $r < \bar{r} \leqslant \tilde{b}_k$. To prove (c) it suffices to show that $r > \tilde{b}_{k-1}$. By (a), $d_{w_{\bar{r}}}(F_{w_{\bar{r}}}) < d_{w_{\tilde{b}_{k-1}}}(F_{w_{\tilde{b}_{k-1}}})$, and hence $\bar{r} > \tilde{b}_{k-1}$. So, for each $e \in F_{w_{\bar{r}}} \backslash E^-$, we have

$$\begin{aligned} w_{\tilde{b}_{k-1}}(e) &= \begin{cases} w(e) + b(e) & \text{for } e \in S, \\ w(e) - b(e) & \text{for } e \in \bar{S}, \end{cases} \\ &= w_{\bar{r}}(e). \end{aligned}$$

Note that $d_{w_{\tilde{b}_{k-1}}}(F_{w_{\bar{r}}}) \geqslant d_{w_{\tilde{b}_{k-1}}}(F_{w_{\tilde{b}_{k-1}}}) > c$. So,

$$\begin{aligned} d_{w_{\bar{r}}}(F_{w_{\bar{r}}} \backslash E^-) + d_{w_{\tilde{b}_{k-1}}}(F_{w_{\bar{r}}} \cap E^-) &= d_{w_{\tilde{b}_{k-1}}}(F_{w_{\bar{r}}} \backslash E^-) + d_{w_{\tilde{b}_{k-1}}}(F_{w_{\bar{r}}} \cap E^-) \\ &= d_{w_{\tilde{b}_{k-1}}}(F_{w_{\bar{r}}}) > c. \end{aligned} \tag{2}$$

On the other hand, by the definition of δ, we have

$$d_{w_{\bar{r}}}(F_{w_{\bar{r}}}) + \mu(F_{w_{\bar{r}}} \cap E^-)\delta = c. \tag{3}$$

For each $e \in E^-$, as $\tilde{b}(e) \geqslant \tilde{b}_k \geqslant \bar{r} > r$, we have

$$w_{\bar{r}}(e) = \begin{cases} w_r(e) + \delta\mu(e) & \text{if } e \in S, \\ w_r(e) - \delta\mu(e) & \text{if } e \in \bar{S}. \end{cases} \tag{4}$$

So,

$$d_{ws_{\bar{r}}}(F_{w_{\bar{r}}} \cap E^-) + \mu(F_{w_{\bar{r}}} \cap E^-)\delta = d_{w_r}(F_{w_{\bar{r}}} \cap E^-). \tag{5}$$

By (2), (3), and (5), we get

$$d_{w_{\tilde{b}_{k-1}}}(F_{w_{\bar{r}}} \cap E^-) > d_{w_r}(F_{w_{\bar{r}}} \cap E^-). \tag{6}$$

We obtain from (6) that

$$r > \tilde{b}_{k-1}. \tag{7}$$

So, (c) holds.

Moreover, since $r > \tilde{b}_{k-1}$, for each $e \in F_{w_{\bar{r}}} \backslash E^-$, we know that $\tilde{b}(e) < r < \bar{r}$, and hence $w_r(e) = w_{\bar{r}}(e)$. Using this result and Equations (3) and (5), we obtain

$$\begin{aligned} d_{w_r}(F_{w_{\bar{r}}}) &= d_{w_r}(F_{w_{\bar{r}}} \backslash E^-) + d_{w_r}(F_{w_{\bar{r}}} \cap E^-) \\ &= d_{w_r}(F_{w_{\bar{r}}} \backslash E^-) + c - d_{w_{\bar{r}}}(F_{w_{\bar{r}}} \backslash E^-) \\ &= c. \end{aligned} \tag{8}$$

Therefore,

$$d_{w_r}(F_{w_r}) \leqslant d_{w_r}(F_{w_{\bar{r}}}) = c, \tag{9}$$

i.e., (d) is also true.

We now prove (a) and (b) for the next iteration from (c) and (d). If Step 1 follows Step 3 of the previous iteration, i.e., $\bar{r}$ is equal to the last r, then by (9) and Step 2 we have $d_{w_{\bar{r}}}(F_{w_{\bar{r}}}) < c$, that is, (a) holds for the $\bar{r}$. Also, by (7) we have $\tilde{b}_k > \bar{r} > \tilde{b}_{k-1}$. Using the same arguments shown in Theorem 2.1, we can deduce (b) for this $\bar{r}$. So, we proved all four conclusions and hence the algorithm is well-defined.

Second, we claim that the algorithm has the following two properties:

(e) $\frac{c-d_{w_r}(F_{w_r})}{c-d_{w_{\bar{r}}}(F_{w_{\bar{r}}})} + \frac{\mu(F_{w_r}\cap E^-)}{\mu(F_{w_{\bar{r}}}\cap E^-)} \leqslant 1$, and

(f) $\mu(F_{w_r} \cap E^-) < \mu(F_{w_{\bar{r}}} \cap E^-)$ and $d_{w_r}(F_{w_r}) > d_{w_{\bar{r}}}(F_{w_{\bar{r}}})$.

In fact, $c - d_{w_{\bar{r}}}(F_{w_{\bar{r}}}) \geqslant c - d_{w_{\bar{r}}}(F_{w_r}) = c - d_{w_r}(F_{w_r}) + d_{w_r}(F_{w_r}) - d_{w_{\bar{r}}}(F_{w_r}) = c - d_{w_r}(F_{w_r}) + \mu(F_{w_r} \cap E^-)\delta = c - d_{w_r}(F_{w_r}) + \mu(F_{w_r} \cap E^-)\frac{c-d_{w_{\bar{r}}}(F_{w_{\bar{r}}})}{\mu(F_{w_{\bar{r}}}\cap E^-)}$. Therefore (e) is proved.

(f) follows (a), (d), and (e) directly.

Note that in [10], the problem is reduced to finding a w such that

$$\max_{F \in \mathcal{F}} \{d_w(F) - \lambda b(F)\} = c,$$

where $b(F)$ is similar to $\mu(F)$ in this paper. When $\lambda = \lambda_k$, the algorithm of [10] solves the problem

$$\max_{F \in \mathcal{F}} \{d_w(F) - \lambda_k b(F)\},$$

obtains the optimal solution and the optimal value F_k and h_k, respectively, and then updates λ_k to obtain λ_{k+1}. In their convergence proof, the authors of [10] first showed that their iterative sequences $\{\lambda_k\}, \{F_k\}$, and $\{h_k\}$ have the following five properties:

(i) $h_k > c$,
(ii) $\lambda_{k+1} - \lambda_k = \frac{h_k - c}{b(F_k)}$,
(iii) $\lambda_{k+1} > \lambda_k$,
(iv) $\frac{h_{k+1} - c}{h_k - c} + \frac{b(F_{k+1})}{b(F_k)} \leqslant 1$,
(v) $b(F_{k+1}) < b(F_k)$ and $h_{k+1} < h_k$.

These five properties are in fact just what Radzik [8] required for proving the finite convergence of his method. Therefore, Zhang and Liu obtained their convergence theorem immediately after establishing the above five properties.

Using our results (a)–(f), it is easy to see that Phase 2 of our algorithm also has the five properties (corresponding to (i)–(v) of [10]). In fact, $d_{w_{\bar{r}}}(F_{w_{\bar{r}}}) < c$ corresponds to (i); $\delta (= \bar{r} - r) = \frac{c - d_{w_{\bar{r}}}(F_{w_{\bar{r}}})}{\mu(F_{w_{\bar{r}}} \cap E^-)}$ corresponds to (ii); $d_{w_{\bar{r}}}(F_{w_{\bar{r}}}) < c$ together with $F_{w_{\bar{r}}} \cap E^- \neq \emptyset$ imply that $\delta > 0$, which corresponds to (iii); $\frac{c - d_{w_r}(F_{w_r})}{c - d_{w_{\bar{r}}}(F_{w_{\bar{r}}})} + \frac{\mu(F_{w_r} \cap E^-)}{\mu(F_{w_{\bar{r}}} \cap E^-)} \leqslant 1$ corresponds to (iv); finally, $\mu(F_{w_r} \cap E^-) < \mu(F_{w_{\bar{r}}} \cap E^-)$ and $d_{w_r}(F_{w_r}) > d_{w_{\bar{r}}}(F_{w_{\bar{r}}})$ correspond to $b(F_{k+1}) < b(F_k)$ and $h_{k+1} < h_k$, i.e. (v). The only difference is that the subproblem of [10] is a maximization problem, while in this paper the subproblem is a minimization problem, and hence the corresponding inequalities and expressions change directions.

So, following the argument in [10], we know that Radzik's proof for convergence is still valid in our case, i.e., the algorithm terminates at an r which meets the equation $d_{w_r}(F_{w_r}) = c$, and hence is the optimal solution of the (IP).

Combining the above analysis, we can conclude that

THEOREM 2.2. *The algorithm in Phase* 2 *terminates after a finite number of iterations with an optimal solution for the (IP). The number of iterations is the same as the number of iterations of Radzik's algorithm.*

Note that Theorem 2.2 can be proved directly. But the above reasoning shows the close relationship between our algorithm and the Radzik's method. Radzik gave an estimate to the required number of iterations for his method, and in Section 3, we will give an improved estimate to Radzik's algorithm. Of course this estimate for the number of iterations is also available to our algorithm due to the second conclusion of the above theorem.

3. An Improved Bound for a Newton Type Method to Solve LFCO Problems

In [8], Radzik considered the class of LFCO problems. An LFCO problem $\mathcal{L}$ can be expressed as

$$\mathcal{L}: \qquad \max \frac{a(1)x(1)+a(2)x(2)+\cdots+a(p)x(p)}{b(1)x(1)+b(2)x(2)+\cdots+b(p)x(p)}, \tag{10}$$

$$\text{subject to} \quad x=(x(1),x(2),\ldots,x(p))\in\mathcal{X}, \tag{11}$$

where $a=(a(1),a(2),\ldots,a(p))$, $b=(b(1),b(2),\ldots,b(p))$, and $\mathcal{X}\subseteq\{0,1\}^p$.

The problem bears the following meaning. Suppose we consider a ground set E of p elements $\{e_1,e_2,\ldots,e_p\}$, and for each subset F of E, define $x_F(e)=1$ if element $e\in F$ and $x_F(e)=0$ otherwise, and call x_F the characteristic vector of F. Furthermore, let $a(i)$ and $b(i)$ be the profit and weight of element e_i, respectively, and express the inner product $c(1)z(1)+c(2)z(2)+\cdots+c(p)z(p)$ of two vectors $c=(c(1),c(2),\ldots,c(p))$ and $z=(z(1),z(2),\ldots,z(p))$ by cz. Then, numbers ax_F, bx_F, and ax_F/bx_F are, respectively, the profit, weight, and mean-weight profit of the subset F. Under such a setting, problem $\mathcal{L}$ becomes to find a subset F that maximizes the mean-weight profit.

In problem $\mathcal{L}$, we assume that $ax>0$ for some $x\in\mathcal{X}$, and $bx>0$ for all $x\in\mathcal{X}$, but there may be some vector $x\in\mathcal{X}$ with non-positive product ax and some components of b with non-positive $b(i)$. Problem $\mathcal{L}$ can be equivalently formulated in the following way.

$$\mathcal{P}: \qquad \text{minimize } \delta, \quad \text{subject to } (ax)-\delta(bx)\leqslant 0 \quad \text{for all } x\in\mathcal{X}. \tag{12}$$

We call $\mathcal{P}$ the parametric version of $\mathcal{L}$ and let δ^* be the optimal solution (and optimal value) to problem $\mathcal{P}$. If we define

$$\begin{aligned} h(\delta) &= \max\{(ax)-\delta(bx) \mid x\in\mathcal{X}\} \\ &= \max\{(a-\delta b)x \mid x\in\mathcal{X}\}, \end{aligned}$$

then we have another equivalent formulation for problem $\mathcal{L}$:

$$\mathcal{R}: \qquad \text{solve} \quad h(\delta)=0. \tag{13}$$

Function $h(\delta)$ is convex, piecewise linear, and decreasing, and the optimal solution δ^* to problem $\mathcal{P}$ is its only root. Note that due to (1), our problem (IP) in Section 2 can also be expressed as a root-finding problem for a combinatorial equation $\bar{h}(r)=c$, where

$$\bar{h}(r) = \min\{w_r x_F \mid x_F \in X_{\mathcal{F}}\},$$

and vector w_r was defined before,

$$x_F(e) = \begin{cases} 1, & e \in F \cap \bar{S}, \\ -1, & e \in F \cap S, \\ 0, & \text{otherwise} \end{cases}$$

and $X_{\mathcal{F}}$ is the set of characteristic vectors x_F for all $F \in \mathcal{F}$. So, the two problems are closely related in nature.

Radzik [8] proposed a method for computing the root δ^* of $h(\delta)$. It can be described formally as follows.
Radzik's algorithm for LFCO:

Step 0: Set $\overline{\delta} = 0$.
Step 1: Maximize the linear function $(a - \overline{\delta} b)x$ over $\mathcal{X}$ and obtain the maximizer $\bar{x}$.
Step 2: If $h(\overline{\delta}) = 0$, then the root $\delta^* = \overline{\delta}$ and the algorithm terminates.
Step 3: Otherwise let $\overline{\delta} = a\overline{x}/b\overline{x}$, go to Step 1.

The method is in fact a discrete type Newton method because if $h(\overline{\delta}) \neq 0$, then we use the following idea to find $\overline{\delta} + \triangle\delta$ which is an approximate root to the equation $h(\delta) = 0$:

$$\begin{aligned} h(\overline{\delta} + \triangle\delta) &= \max_{x \in X}(a - (\overline{\delta} + \triangle\delta)b)x \\ &\approx (a - (\overline{\delta} + \triangle\delta)b)\bar{x} \\ &= 0, \end{aligned} \tag{14}$$

i.e.,

$$\overline{\delta} + \triangle\delta = \frac{a\bar{x}}{b\bar{x}}$$

and let this $\overline{\delta} + \triangle\delta$ be the next iterative point. Note that in (14) we suppose the maximum is reached at $\bar{x}$ because it is true when $\delta = \overline{\delta}$, and we assume $\triangle\delta$ is small.

Note that by the same idea but with a bit more complicated analysis, we can see that our algorithm in Phase 2 is in fact also a discrete type Newton algorithm.

Next, we prove that Radzik's algorithm has a better complexity bound than what is given in [8].

Let δ_i be the value of $\overline{\delta}$ at the beginning of the ith iteration, and x_i, h_i, f_i, and g_i be, respectively, vector $\overline{x}$ and scalars $(a - \delta_i b)\overline{x}$, $a\overline{x}$, and $b\overline{x}$ from this iteration. Thus

$$\begin{aligned} h_i &= (a-\delta_i b)x_i = \max\{(a-\delta_i b)x \mid x \in \mathcal{X}\} \\ &= f_i - \delta_i g_i, \\ \delta_{i+1} &= \frac{ax_i}{bx_i} = \frac{f_i}{g_i}. \end{aligned} \tag{15}$$

It is easy to see that

$$\delta_{i+1} - \delta_i = \frac{h_i}{g_i}.$$

[8] also proved that

$$\frac{h_{i+1}}{h_i} + \frac{g_{i+1}}{g_i} \leqslant 1$$

and the iterations terminate finitely. Let t be the index of the last iteration. Then it was shown in [8] that

$$h_1 > h_2 > \cdots > h_{t-1} > h_t = 0,$$

$$0 = \delta_1 < \delta_2 < \cdots < \delta_{t-1} < \delta_t = \delta^*,$$

$$g_1 > g_2 > \cdots > g_{t-1} \geqslant g_t.$$

Radzik proved the following complexity bound for the Newton type method.

THEOREM 3.1 ([8]). *The Newton type method applied to an LFCO problem finds the optimal solution in $O(p^2 \log^2 p)$ iterations.*

We find that actually the above complexity bound for the Newton type method can be improved. Our improvement is based on a result given by the next theorem which is due to Goemans in a private communication with Radzik [8].

THEOREM 3.2 ([8]). *Let $c = (c(1), c(2), \ldots, c(p))$ be a vector with non-negative real coordinates. Let $y_1, y_2, \ldots, y_q$ be vectors from $\{-1, 0, 1\}^p$. If for all $i = 1, 2, \ldots, q-1$,*

$$0 < y_{i+1}c \leqslant \tfrac{1}{2} y_i c,$$

then $q = O(p \log p)$.

Below we first use this theorem to estimate the number of times that make $h_{k+1} \geqslant (1/2)h_k$ in using the Newton type method for solving the LFCO problem.

THEOREM 3.3 *When the problem $\mathcal{R}$ is solved by the Newton type method, there are at most $O(p \log p)$ iterations k such that $h_{k+1} \geqslant (1/2)h_k$.*

Proof. Let $k_1 < k_2 < \cdots < k_q$ be all indices of k such that $h_{k+1} \geqslant (1/2)h_k$. As for each k,

$$\frac{h_{k+1}}{h_k} + \frac{g_{k+1}}{g_k} \leqslant 1,$$

we have

$$0 < g_{k_i+1} \leqslant \tfrac{1}{2} g_{k_i}. \tag{16}$$

As $\{g_i\}$ is decreasing, $g_{k_{i+1}} \leqslant g_{k_i+1}$, and hence (16) means

$$0 < g_{k_{i+1}} \leqslant \tfrac{1}{2} g_{k_i},$$

i.e.,

$$0 < bx_{k_{i+1}} \leqslant \tfrac{1}{2} bx_{k_i}. \tag{17}$$

If we define vector $c = (c(1), c(2), \ldots, c(p))$ as

$$c(j) = |b(j)|$$

for $j = 1, 2, \ldots, p$, and define vectors y_i $(i = 1, 2, \ldots, q)$ as

$$y_i(j) = \text{sign}(b(j)) \cdot x_{k_i}(j)$$

for $j = 1, 2, \ldots, p$, then c must be a non-negative vector and each vector $y_i \in \{-1, 0, 1\}^p$. Moreover,

$$bx_{k_i} = y_i c, \quad i = 1, 2, \ldots, q.$$

So (17) means

$$0 < y_{i+1} c \leqslant \tfrac{1}{2} y_i c.$$

Now by Theorem 3.2 we know that $q = O(p \log p)$. The proof is completed. □

We now estimate the number of times to have $h_{k+1} \leqslant (1/2)h_k$.

THEOREM 3.4 *When the problem $\mathcal{R}$ is solved by the Newton type method, there are at most $O(p^2 \log p)$ iterations k such that $h_{k+1} \leqslant (1/2)h_k$.*

Proof. Let $k_1 < k_2 < \cdots < k_q$ be the indices of k such that $h_{k+1} \leqslant (1/2)h_k$. We have

$$h_i = f_i - \delta_i g_i = f_i - \frac{f_{i-1}}{g_{i-1}} g_i.$$

As $\{h_i\}$ is decreasing, we know that

$$h_{k_{i+1}} \leqslant h_{k_i+1} \leqslant \tfrac{1}{2} h_{k_i},$$

which means

$$f_{k_{i+1}} - \frac{f_{k_{i+1}-1}}{g_{k_{i+1}-1}} g_{k_{i+1}} \leqslant \frac{1}{2}\left(f_{k_i} - \frac{f_{k_i-1}}{g_{k_i-1}} g_{k_i} \right). \tag{18}$$

As $\{g_i\}$ is decreasing, $g_{k_i-1} > g_{k_{i+1}-1} > 0$. So, from (18) we obtain

$$f_{k_{i+1}} g_{k_{i+1}-1} - f_{k_{i+1}-1} g_{k_{i+1}} \leqslant \tfrac{1}{2}(f_{k_i} g_{k_i-1} - f_{k_i-1} g_{k_i}).$$

Putting $s_i = f_{k_i} g_{k_i-1} - f_{k_i-1} g_{k_i}$, we have that for each $i = 1, 2, \ldots, q$,

$$0 < s_{i+1} \leqslant \tfrac{1}{2} s_i.$$

Let

$$\Lambda_i = \{ j \mid x_{k_i}(j) = 1 \}$$

and

$$\Lambda_i' = \{ j \mid x_{k_i-1}(j) = 1 \}.$$

Then

$$\begin{aligned} s_i &= a x_{k_i} b x_{k_i-1} - a x_{k_i-1} b x_{k_i} \\ &= \left(\sum_{j \in \Lambda_i} a(j) \right) \left(\sum_{l \in \Lambda_i'} b(l) \right) - \left(\sum_{l \in \Lambda_i'} a(l) \right) \left(\sum_{j \in \Lambda_i} b(j) \right) \\ &= \sum_{j,l=1}^{p} z_{jl} a(j) b(l) + \sum_{j,l=1}^{p} z_{lj}' a(l) b(j), \end{aligned}$$

where $z_{jl}=1$ and $z'_{lj}=-1$ if $j\in\Lambda_i$ and $l\in\Lambda'_i$, and $z_{jl}=z'_{lj}=0$ otherwise. We choose c as a $2p^2$-dimensional vector:

$$(|a(1)b(1)|,|a(1)b(2)|,\dots,|a(1)b(p)|,|a(2)b(1)|,|a(2)b(2)|,\dots,|a(p)b(p)|,\\ |a(1)b(1)|,|a(1)b(2)|,\dots,|a(1)b(p)|,|a(2)b(1)|,|a(2)b(2)|,\dots,|a(p)b(p)|)$$

and define a $2p^2$-dimensional vector y_i for each $i=1,2,\dots,q$, such that the first p^2 components of y_i are equal to $\text{sign}(a(j)b(l))\cdot z_{jl}$ and the last p^2 components are equal to $\text{sign}(a(l)b(j))\cdot z'_{lj}$. In this way each y_i is a vector consisting of only three possible values 0, -1, and 1, and $s_i=y_ic$, for each $i=1,2,\dots,q$. Applying Theorem 3.2, we conclude that $q=O(2p^2\log(2p^2))=O(p^2\log p)$. □

Combining the above two theorems, it is immediate to obtain the following theorem that improves Radzik's estimate.

THEOREM 3.5 *The Newton type method applied to an LFCO problem finds the optimal solution in $O(p^2\log p)$ iterations.*

As explained before, this estimate of time complexity is also available to our method in Section 2.

4. Conclusion

In this paper, we propose a weakly dominant relationship and study an inverse model to make a given set become a weakly dominant set. Under weighted l_∞ norm, we prove that the inverse model can be solved by a Newton type method which has the same convergent properties as what Radzik's algorithm has for LFCO problems. We also present an improved complexity bound for Radzik's algorithm.

In the end, we like to note that under the ordinary l_∞ norm in our inverse model, i.e. all $p(e)=1$, it is not difficult to show that our algorithm in Phase 2 needs at most $O(\max\{|F|\mid F\in\mathcal{F}\})$ iterations, and the total complexity to solve the model is $O(\mathcal{A}(\log(E)+\max\{|F|\mid F\in\mathcal{F}\}))$.

Acknowledgement

The authors thank Professor M. Cai (the Science Academy of China) for his very helpful suggestion in revising the paper.

References

1. Burton, D., Pulleyblank, W. R., and Toint, Ph. L. (1997), The inverse shortest paths problem with upper bounds on shortest paths costs, *Network Optimization,* (Gainesville, FL, 1996), Springer, Berlin, 156–171.

2. Burton, D. and Toint, Ph. L. (1992), On an instance of the inverse shortest paths problem, *Mathematical Programming,* 53, 45–61.
3. Fekete, S. P., Hochstattler, W., Kromberg, St. and Moll, Ch. (1999), The complexity of an inverse shortest paths problem, *Comtemporary Trends in Discrete Mathematics,* American Mathematical Society, Providence, RI, 113–127.
4. Fredman, M. L. and Tarjan, R. E. (1987), Fibonacci heaps and their uses in improved network optimization algorithms, *Journal of the ACM,* 34, 596–615.
5. Heuberger, C. (2004), Inverse optimization, a survey on problems, methods, and results, *Journal of Combinatorial Optimization*, 8, 329–361.
6. Megiddo, N. (1983), Applying parallel computation algorithms in the design of serial algorithms, *Journal of the Association on Computing Machinery,* 30, 852–865.
7. Megiddo, N. (1979), Combinatorial optimization with rational objective functions, *Mathematics of Operations Research,* 4, 414–424.
8. Radzik, T. (1993), Parametric flows, weighted means of cuts, and fractional combinatorial optimization. In Pardalos P.M. (ed.) *Complexity in Numerical Optimization*, World Scientific Publishing Co, pp. 351–386.
9. Zhang, J. Z. and Lin, Y. X. (2003), Computation of the reverse shortest-path problem, *Journal of Global Optimization*, 25, 243–261.
10. Zhang, J. and Liu, Z. (2002), A general model of some inverse combinatorial optimization problems and its solution method under l_∞ norm. In *Journal of Combinatorial Optimization*, 6, 207–227.
11. Zhang, J. and Liu, Z. (2002), An oracle-strongly polynomial algorithm for bottleneck expansion problems. In *Optimization Methods and Software*, 17, 61–75.