Inverse Optimization, Part I: Linear Programming and General Problem

by

Ravindra K. Ahuja
James B. Orlin

SWP# 4002 February 1998

Inverse Optimization, Part 1:
Linear Programming and General Problem

Ravindra K. Ahuja*
Sloan School of Management
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

James B. Orlin
Sloan School of Management
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

(Revised January 27, 1998)

* On leave from Indian Institute of Technology, Kanpur 208 016, INDIA.

Inverse Optimization, Part 1: Linear Programming and General Problem

Ravindra K. Ahuja[1] and James B. Orlin[2]

ABSTRACT

In this paper, we study inverse optimization problems defined as follows: Let **S** denote the set of feasible solutions of an optimization problem **P**, let c be a specified cost vector, and x^0 be a given feasible solution. The solution x^0 may or may not be an optimal solution of **P** with respect to the cost vector c. The inverse optimization problem is to perturb the cost vector c to d so that x^0 is an optimal solution of **P** with respect to d and $||d - c||_p$ is minimum, where $||d - c||_p$ is some selected L_p norm. In this paper, we consider the inverse linear programming problem under the L_1 norm (where we minimize $\sum_{j\in J}|d_j - c_j|$, with J denoting the index set of variables x_j) and under the L_∞ norm (where we minimize $\max\{|d_j - c_j|: j \in J\}$). We show that the dual of the inverse linear programming problem with the L_1 norm reduces to a modification of the original problem obtained by eliminating the non-binding constraints (with respect to x^0) and imposing the following additional lower and upper bound constraints: $|x_j - x_j^0| \le 1$ for all $j \in J$. We next study the inverse linear programming problem with the L_∞ norm and show that its dual reduces to a modification of the original problem obtained by eliminating the non-binding constraints (with respect to x^0) and imposing the following single additional constraint: $\sum_{j\in J}|x_j - x_j^0| \le 1$. Finally, we show that (under reasonable regularity conditions) if the problem **P** is polynomially solvable then the inverse versions of **P** under L_1 and L_∞ norms are also polynomially solvable. This result uses ideas from the ellipsoid algorithm and, therefore, does not lead to combinatorial algorithms for solving inverse optimization problems.

[1] Sloan School of Management, MIT, Cambridge, MA 02139, USA; On leave from Indian Institute of Technology, Kanpur 208 016, INDIA.

[2] Sloan School of Management, MIT, Cambridge, MA 02139, USA.

1. INTRODUCTION

Inverse problems have been studied extensively by researchers working with geophysical data. Tarantola [1987] describes inverse problems in the following manner: "*Let S represent a physical system. Assume that we are able to define a set of model parameters which completely describe S. These parameters may not all be directly measurable (such as the radius of Earth's metallic core). We can operationally define some observable parameters whose actual values hopefully depend on the values of the model parameters. To solve the forward problem is to predict the values of the observable parameters, given arbitrary values of the model parameters. To solve the inverse problem is to infer the values of the model parameters from given observed values of the observable parameters.*"

In terms of the above notation, a typical optimization problem is a forward problem since it identifies the values of observable parameters (decision variables) given the values of the model parameters (cost coefficients, right-hand side vector, and the constraint matrix). An inverse optimization problem consists of inferring the values of the model parameters (cost coefficients, right-hand side vector, and the constraint matrix) given the values of observable parameters (decision variables). In the past few years, there has been sufficient interest in inverse optimization problems in the operations research community, and a variety of inverse optimization problems have been studied by researchers.

In this paper, we study inverse optimization problems defined in the following manner. Let $\mathbf{P}$ denote an (instance of an) optimization problem with $\mathbf{S}$ as the set of feasible solutions and c as the cost vector; that is, $\mathbf{P} = \min\{cx : x \in \mathbf{S}\}$. Suppose that $x^0 \in \mathbf{S}$. The solution x^0 may or may not be an optimal solution of $\mathbf{P}$ with respect to the cost vector c. For a cost vector d, we define $\mathbf{P}(d)$ as a variation of problem $\mathbf{P}$ with the cost vector c replaced by d, that is, $\mathbf{P}(d) = \min\{dx : x \in \mathbf{S}\}$. An inverse optimization problem with L_p norm is to identify a cost vector d such that x^0 is an optimal solution of $\mathbf{P}(d)$ and $||d - c||_p = [\sum_{j \in J} |d_j - c_j|^p]^{1/p}$ is minimum, where J denotes the index set of variables x_j. In words, the inverse optimization problem is to perturb the cost vector c to d so that x^0 is an optimal solution with respect to the perturbed cost vector and the cost of perturbation is minimum. In Section 2, we describe several applications of the inverse optimization problems and give references for some other applications.

We briefly survey the available research on inverse optimization problems. Geophysical scientists were the first ones to study inverse problems. The book by Tarantola [1987] gives a comprehensive discussion of the theory of inverse problems in the geophysical sciences. Within the mathematical programming community, the interest in inverse optimization problems was generated by the papers due to Burton and Toint [1992, 1994] who studied inverse shortest path problems arising in seismic tomography used in predicting the movement of earthquakes. In the past few years, inverse optimization problems have been studied rather intensively. The table shown in Figure 1 summarizes the references on inverse optimization of relevance to mathematical programmers.

Inverse shortest path problem	Burton and Toint [1992, 1994], Burton, Pulleyblank, and Toint [1997], Cai and Yang [1994], Xu and Zhang [1995], Zhang, Ma, and Yang [1995], and Dial [1997]
Inverse maximum capacity path problem	Yang and Zhang [1996]
Inverse spanning tree problem	Ma, Xu, and Zhang [1995], Sokkalingam, Ahuja, and Orlin [1995], and Ahuja and Orlin[1998a]
Inverse sorting problem	Ahuja and Orlin [1997]
Inverse shortest arborescence problem	Hu and Liu [1995]
Inverse bipartite k-matching problem	Huang and Liu [1995a]
Inverse minimum cut problem	Yang, Zhang, and Ma [1997], and Zhang and Cai [1998]
Inverse minimum cost flow problem	Huang and Liu [1995b], and Sokkalingam [1996]
Inverse matroid intersection problem	Cai and Li [1995]
Inverse polymatroidal flow problem	Cai, Yang, and Li [1996]

Figure 1. Reference on inverse optimization problems.

We will now briefly survey our research on inverse optimization. In Sokkalingam, Ahuja and Orlin [1996], we studied the inverse spanning tree problem and developed an $O(n^3)$ algorithm under the L_1 norm and an $O(n^2)$ algorithm under the L_∞ norm, where n is the number of nodes in the network. Subsequently, Ahuja and Orlin [1998a] suggested an $O(n^2 \log n)$ algorithm to solve the inverse spanning tree problem under L_1 norm. Ahuja and Orlin [1997] studied the convex ordered set problem, a generalization of the inverse sorting problem.

In this paper, we consider inverse optimization problems under the L_1 and L_∞ norms. We first consider inverse linear programming problem under the L_1 norm (that is, we minimize $\sum_{j \in J} |d_j - c_j|$) and the L_∞ norm (that is, we minimize $\max\{|d_j - c_j| : j \in J\}$. Finally, we consider general inverse optimization problems under L_1 and L_∞ norms. The second part of this paper, Ahuja and Orlin [1997b] consider the following special cases of the inverse linear programming problem under the L_1 and L_∞ norms: the shortest path problem, the assignment problem, the minimum cut problem, and the minimum cost flow problem. In an another paper, Ahuja and Orlin [1997c] consider inverse network flow problems for the unit weight case as in the second part of the paper and develop combinatorial proofs that do not rely on the inverse linear programming theory.

The major contributions made in this paper are as follows:

1. We show that if the problem **P** is a linear programming problem, then its inverse problem under the L_1 norm is also a linear programming problem. The dual of the inverse problem has the same objective function as **P** and the constraint set comprises only the binding constraints of **P** (with respect to the solution x^0) plus the following additional lower and upper bound constraints on the variables: $x_j^0 - 1 \le x_j \le x_j^0 + 1$ for all j, $1 \le j \le n$ (or, alternatively, $|x_j - x_j^0| \le 1$, for all j, $1 \le j \le n$).

2. We show that if the problem **P** is a linear programming problem, then its inverse problem under the L_∞ norm is also a linear programming problem. The dual of the inverse problem has the same objective function as **P**, and the constraint set comprises only the binding constraints of **P** (with respect to the solution x^0) plus the following additional constraint: $|x_j - x_j^0| \le 1$.

3. We also study the weighted versions of L_1 and L_∞ norms, where the objective functions is to minimize $\sum_{j \in J} w_j |d_j - c_j|$, and to minimize $\max\{w_j|d_j - c_j| : j \in J\}$, respectively, for some non-negative weights w_j's.

4. We present a unified approach to solve inverse linear programming problems. When the linear programming results are adapted for special cases, such as network flow problems, we match or improve many results obtained by several researchers.

5. We show that (under reasonable regularity conditions) if the problem **P** is polynomially solvable, then inverse versions of **P** under the L_1 and L_∞ norms are also polynomially solvable. This result uses ideas from ellipsoid algorithm and, therefore, does not lead to combinatorial algorithms for solving inverse optimization problems.

2. Applications of Inverse Optimization Problems

In this section, we briefly describe several applications of inverse optimization problems collected from the literature and provide references for a few other applications.

Geophysical Sciences

Geophysical scientists often do not have all the model parameters, since they may be very difficult or impossible to determine (such as the radius of Earth's metallic core). They may have some estimates of model parameters and values of the observable parameters are used to improve the estimates of the model parameters. Consequently, inverse problems have been extensively studied by geophysical scientists (see, for example, Neumann-Denzau and Behrens [1984], Nolet [1987], Tarantola [1987], and Woodhouse and Dziewonski [1984]). An important application in this area concerns predicting the movements of earthquakes. To model earthquake movements, consider a network obtained by the discretization of a geologic zone into a number of square cells. Nodes corresponding to adjacent cells are connected by arcs. The cost of an arc represents the transmission time of certain seismic waves from corresponding cells, and is not accurately known. Earthquakes are then observed and the arrival times of the resulting seismic perturbations at various observation stations are observed. Assuming that the earthquakes travel along shortest paths, the problem faced by geologists is to reconstruct the transmission times between cells from the observation of shortest time waves and a priori knowledge of the geologic nature of the zone under study. This problem is an example of an inverse shortest path problem. Inverse problems also arise in X-ray tomography where observations from a CT-scan of a body part together with a priori knowledge of the body is used to estimate its dimension. The book by Tarantola [1987] gives a comprehensive treatment of the theory of inverse problems and provides additional applications.

Isotonic Regression

An important application of inverse problem arises in isotonic regression. The *isotonic regression problem* is defined as follows: Given a = $\{a_1, a_2, \dots, a_n\} \in R^n$, find x = $\{x_1, x_2, \dots, x_n\} \in R^n$, so as to minimize $||x - a||_p = [\sum_{j=1}^{n} |x_j - a_j|^p]^{1/p}$ for some positive integer p subject to the *isotonicity* (or *monotonicity*) constraints $x_1 \le x_2 \le \dots \le x_n$. The isotonic regression problem is an important problem in regression. The isotonic regression problem arises in statistics, production planning, and inventory control; the books by Barlow et al. [1972] and Robertson et al. [1988] describe several applications. As an application of isotonic regression, consider a fuel tank where fuel is being consumed at a slow pace and measurements of the fuel tank are taken at different points in time. Suppose that these measurements are $a_1, a_2, \dots, a_n$. Due to the errors in the measurements, these numbers may not be in the non-increasing order despite the fact that that the true amounts of fuel remaining in the tank are non-increasing. However, we need to determine these measurements as accurately as possible. One possible way to accomplish this could be to perturb these numbers to are $x_1, x_2, \dots, x_n$ so that $x_1 \ge x_2 \ge \dots \ge x_n$ and the cost of perturbation given by $C_1(x_1-a_1) + C_2(x_2-a_2) + \dots + C_n(x_n-a_n)$ is minimum, where $C_j(x_j)$'s are convex functions that give the cost of perturbing the data. (This problem may be transformed to the isotonic regression problem by replacing x_j's by their negatives.) This is clearly an example of an inverse problem where a priori knowledge about the system (that the observations must be in the non-increasing order) is used together with the observations ($a_1, a_2, \dots, a_n$) to estimate the model parameters ($x_1, x_2, \dots, x_n$). Ahuja and Orlin [1997] describe highly efficient algorithms to solve the isotonic regression problem, which they also refer to as the convex ordered set problem.

Traffic Equilibrium

In a transportation network, users make a number of trips between different origin-destination pairs. Travel costs are flow dependent and as the flow increases so does the travel costs. Drivers usually select their routes in a way so as to minimize their travel cost (or time). Under certain idealized assumptions, the resulting flow in such a network is a *user equilibrium flow*, where no user can decrease its travel cost unilaterally by changing its route (see, for example, Sheffi [1985]). This user equilibrium flow does not necessarily correspond to the most efficient way of using the transportation network.

A transportation planner may want to enforce a flow that minimizes the total travel cost over the network; such a flow is called the *system optimal flow*. A user equilibrium flow may or may not be the same as the system optimal flow. If not, then tolls may be imposed on some road segments of the route so that the user equilibrium flow becomes identical to the system optimal flow. If we denote by x^0 the system optimal flow, by x^* the user equilibrium flow, then imposing tolls amounts to changing travel costs so that the user equilibrium changes and becomes the same as the system optimal flow x^0. This is an example of the inverse optimization problem. In case, the objective is to impose the minimum total toll to make the user equilibrium flow identical to the system optimal flow, then the resulting problem is an instance of the inverse optimization problem under the L_1 norm. In case, the objective is to minimize the maximum toll imposed on any road, then the resulting problem is an instance of the inverse optimization problem under the L_∞ norm. As a matter of fact, these two problems are instances of the inverse multicommodity flow problem where flow between different origin-destination pairs is treated as a different commodity. This problem has been studied by Burton and Toint [1992, 1994] and Dial [1997].

A Metric for Determining Deviation from Optimality

Consider a difficult optimization problem which, due to its intractability, cannot be solved optimally for reasonably large size instances. Consequently, we use a heuristic method to solve this problem. To assess the quality of a heuristically developed solution, one needs *metrics*, also known as *performance measures*. The most widely accepted performance measure for assessing the quality of a solution is the relative error, given by $[z' - z^*]/z^*$, where z^* is the optimal solution value and z' is a value of the solution obtained by some heuristic method. (In practice, z^* is replaced by a more readily obtainable lower bound on z^*.) We can use ideas from inverse optimization to define an alternative measure of performance that can provide useful insights in practice. We say that x^0 is *ε-optimal* if the cost of each variable can be perturbed by at most ε% of its original value and x^0 can be made optimal for the perturbed cost vector. This inverse perspective is natural in cases where the objective function is only known approximately, as is typically the case in practice.

Multi-Criteria Optimization

Inverse optimization problems have potential applications in multi-criteria optimization and data envelopement analysis. Consider a multi-criteria optimization problem with different objectives $z_1(x)$, $z_2(x)$, ... , $z_k(x)$. One standard technique to solve multi-criteria optimization problem is to reduce it to a single criterion optimization problem by considering a weighted sum of different objectives; suppose that w_1, w_2, ... , w_k is a given set of weights. Now suppose that we are given a proposed solution x^0, and we ask the question how to perturb theses weights by a minimum amount so that the solution x^0 is optimal for the weighted problem. This problem is an example of the inverse optimization problem.

Stability Analysis

Inverse optimization is related to stability analysis which is a type of post-optimality analysis similar to sensitivity analysis. Typically in sensitivity analysis, one parameter (such as the cost of one variable) is allowed to change, but in the stability analysis, multiple parameters are allowed to change simultaneously. Consider an optimization problem **P**(c) with c as the cost vector and let x^0 denote an optimal solution of **P**(c). For a given L_p norm, the *stability problem* is to determine the largest value of ε such that x^0 is optimal for **P**(d) for all values of d satisfying $||d - c||_p \le \varepsilon$. In the inverse problem, we are given a solution x^0 which is not optimal for **P**(c) and we want to determine the smallest value of ε such that x^0 is optimal for **P**(d) for some d satisfying $||d - c||_p \le \varepsilon$. The relationship between the stability problem and the inverse optimization problem can also be explained in geometric terms. Let **D** denote the polyhedron of all cost vectors d such that x^0 is optimal for **P**(d). In the stability problem, we are given a point $c \in$ **D**, and we wish to determine a ball of largest diameter with its center at c and which is fully contained in **D**. In the inverse optimization problem, we are given a point $c \notin$ **D**, and we wish to find the smallest distance between c and the polyhedron **D**. We refer the reader to the survey papers by Sotskov et al. [1995, 1997] and Greenberg [1998] for additional material on stability analysis.

3. Formulating the Inverse Linear Programming Problem

In this section, we study the inverse linear programming problem under the L_1 norm. We will consider the inverse version of the following linear programming problem, which we shall subsequently refer to as **LP**:

$$\text{Minimize } \textstyle\sum_{j \in J} c_j x_j, \tag{3.1a}$$

subject to

$$\sum_{j\in J} a_{ij} x_j \geq b_i, \quad \text{for all } i \in I, \tag{3.1b}$$

where J denotes the index set of the decision vector x, and I denotes the index set of the constraints. Notice that we have not imposed any non-negativity constraints on the variables x. If there are any such constraints, they can be added as explicit constraints and included in the constraint set (3.1b).

Let us associate the dual variable π_i with the i^{th} constraint in (3.1b). The dual of **LP** is the following linear program:

$$\text{Maximize } \sum_{i\in I} b_i\pi_i, \tag{3.2a}$$

subject to

$$\sum_{i\in I} a_{ij} \pi_i = c_j, \quad \text{for all } j \in J, \tag{3.2b}$$

$$\pi_i \geq 0, \quad \text{for all } i \in I. \tag{3.2c}$$

One form of the linear programming optimality conditions states that the solutions x and π are optimal for their respective problems if x is feasible for (3.1), π is feasible for (3.2), and together they satisfy the following condition:

$$\sum_{j\in J} c_j x_j = \sum_{i\in I} b_i \pi_i. \tag{3.3}$$

An alternative to (3.3) is the complementary slackness conditions, which state that if a primal (or, dual) constraint is non-binding, that is, has a positive slack, then the corresponding dual (or, primal) variable must be zero. Alternatively,

$$\text{for any } i \in I, \text{ if } \sum_{j\in J} a_{ij} x_j > b_i \text{ then } \pi_i = 0. \tag{3.4}$$

Let x^0 be a feasible solution of (3.1). We want to make x^0 an optimal solution of (3.1) by perturbing the cost vector c. Recall from Section 3 that we denote by **LP**(d) the linear program (3.1) where the c_j's have been replaced with d_j's. We call d *inverse feasible* with respect to x^0 if x^0 is an optimal solution of **LP**(d). We denote by **INV**(**LP**, x^0, p) the inverse of **LP** with respect to the solution x^0 under the L_p norm. Now notice that

x^0 is an optimal solution of **LP**(d) if and only if there exists a dual solution π that satisfies (3.2b) and (3.2c) with c_j replaced by d_j and the pair (x^0, π) satisfies the complementary slackness conditions. This observation yields the following property that gives a characterization of inverse feasible cost vectors.

Property 1. *A cost vector d is inverse feasible to* **LP** *with respect to the solution* x^0 *if and only if there exists a dual solution* π *satisfying the following conditions:*

C1. The dual solution π *satisfies the conditions in (3.2b) and (3.2c) with* c_j *replaced by* d_j.

C2. The pair (x^0, π) *satisfies the optimality conditions.*

Let **D(LP**, x^0) denote the set of all inverse feasible cost vectors of **LP**. Property 1 together with (3.3) implies that **D(LP**, x^0) consists of all cost vectors d's satisfying the following conditions:

$$\sum_{i\in I} a_{ij}\,\pi_i = d_j, \qquad \text{for all } j \in J, \tag{3.5a}$$

$$\sum_{j\in J} c_j\,x_j^0 = \sum_{i\in I} b_i\,\pi_i, \tag{3.5b}$$

$$\pi_i \geq 0, \qquad \text{for all } i \in I. \tag{3.5c}$$

Alternatively, Property 1 together with (3.4) implies that **D(LP**, x^0) consists of all d's satisfying:

$$\sum_{i\in I} a_{ij}\,\pi_i = d_j, \qquad \text{for all } j \in J, \tag{3.6a}$$

$$\pi_i \geq 0, \qquad \text{for all } i \in I, \tag{3.6b}$$

$$\text{for any } i \in I, \text{ if } \sum_{j\in J} a_{ij}\, x_j^0 > b_i \text{ then } \pi_i = 0. \tag{3.6c}$$

Let B denote the index set of all binding constraints with respect to the solution x^0, that is, the set of all $i \in I$ which satisfy $\sum_{j\in J} a_{ij}\, x_j^0 = b_i$. The conditions in (3.6c) imply that $\pi_i = 0$ for all $i \notin B$. Substituting $\pi_i = 0$ for all $i \notin B$ in (3.6a) and (3.6b) yields the following characterization of **D(LP**, x^0):

$$\sum_{i\in B} a_{ij}\, \pi_i = d_j, \qquad \text{for all } j \in J, \tag{3.7a}$$

$$\pi_i \geq 0, \qquad \text{for all } i \in B. \tag{3.7b}$$

The characterization (3.7) is a more concise characterization of **D**(**LP**, x^0) compared to (3.5); we will thus use (3.7) in the rest of the paper. We summarize the preceding discussion using the following property:

Property 2. *The set* **D**(**LP**, x^0) *of all inverse feasible cost vectors consists of all d satisfying the following constraints:*

$$\sum_{i\in B} a_{ij}\, \pi_i = d_j, \quad \textit{for all } j \in J, \tag{3.8a}$$

$$\pi_i \geq 0, \quad \textit{for all } i \in B, \tag{3.8b}$$

where B is the index set of binding constraints with respect to the solution x^0.

The inverse problem **INV**(**LP**, x^0, p) is to minimize $\|d - c\|_p$ subject to $d \in$ **D**(**LP**, x^0). Since minimizing $[\sum_{j\in J} |d_j - c_j|^p]^{1/p}$ is equivalent to minimizing $\sum_{j\in J} |d_j - c_j|^p$, **INV**(**LP**, x^0, p) can be equivalently stated as follows:

$$\text{Minimize } \sum_{j\in J} |d_j - c_j|^p, \tag{3.9a}$$

subject to

$$\sum_{i\in B} a_{ij}\, \pi_i = d_j, \quad \text{for all } j \in J, \tag{3.9b}$$

$$\pi_i \geq 0, \quad \text{for all } i \in B. \tag{3.9c}$$

Notice that the constraints of the inverse problem **INV**(**LP**, x^0) are closely related to the constraints of the dual of the problem **LP**; they are simply the constraints obtained by changing the right-hand side vector from c to d and setting $\pi_i = 0$ for all $i \notin B$. It is well known that the function $\sum_{j\in J} |d_j - c_j|^p$ is a convex function of the cost vector d. We have therefore shown that the inverse of a linear programming problem is a mathematical programming problem where the objective function is a convex function and the constraints are linear.

We can also impose some additional linear constraints on the cost vector d for specific situations without changing the structure of the mathematical program. For example, in some applications it may be required that the costs can only go up but cannot go down; we can handle this situation by adding constraints of the type $d_j \geq l_j$. In other applications, we may impose upper bounds on the modified cost vector; we can handle this situation by adding constraints of the type $d_j \leq u_j$. We can also solve weighted versions of the inverse problem without changing the structure of the problem, where the objective function is to minimize $\Sigma_{j \in J} w_j|d_j - c_j|^p$.

In the rest of the paper, we will focus on the following objective functions for the inverse problem, all of which may be linearized:

L_1 norm:	minimize $\Sigma_{j \in J}|d_j - c_j|$,
Weighted L_1 norm:	minimize $\Sigma_{j \in J} w_j|d_j - c_j|$,
L_∞ norm:	minimize $\max\{|d_j - c_j| : j \in J\}$,
Weighted L_∞ norm:	minimize $\max\{w_j|d_j - c_j| : j \in J\}$.

4. Solving the Inverse Linear Programming Problem under the L_1 Norm

In this section, we will consider inverse linear programming problems under the L_1 and the weighted L_1 norms. Consider the linear program **LP** given by (3.1) and we wish to solve the inverse problem **INV(LP**, x^0, p) under the L_1 norm. We have shown in the previous section that **INV(LP**, x^0, p) reduces to solving minimize $\Sigma_{j \in J}|d_j - c_j|$, subject to (3.9b) and (3.9c). This is not a linear programming problem in its current form, but can be easily converted to one using a standard transformation. It is well known that minimizing $|d_j - c_j|$ is equivalent to minimizing $\alpha_j + \beta_j$, subject to $d_j - c_j = \alpha_j - \beta_j$, $\alpha_j \geq 0$ and $\beta_j \geq 0$. Using this transformation, the inverse linear programming problem can be stated as follows:

$$\text{Minimize } \Sigma_{j \in J} \alpha_j + \Sigma_{j \in J} \beta_j,$$

or, equivalently,

$$\text{Maximize } -\Sigma_{j \in J} \alpha_j - \Sigma_{j \in J} \beta_j, \tag{4.1a}$$

subject to

$$\sum_{i \in B} a_{ij} \pi_i - \alpha_j + \beta_j = c_j, \text{ for all } j \in J, \tag{4.1b}$$

$$\pi_i \geq 0 \text{ for all } i \in B;\ \alpha_j \geq 0 \text{ and } \beta_j \geq 0 \text{ for all } j \in J. \tag{4.1c}$$

We will now simplify (4.1). We first note that in an optimal solution of (4.1) both of α_j and β_j cannot take positive values, since otherwise we can reduce both of them by a small amount δ without violating any constraint and strictly improving the objective function value. We can restate (4.1) as

$$-\alpha_j + \beta_j = c_j^{\pi}, \text{ for all } j \in J, \tag{4.2}$$

where $c_j^{\pi} = c_j - \sum_{i \in B} a_{ij} \pi_i$. There are three cases to consider.

Case 1. $c_j^{\pi} > 0$. The non-negativity of α_j and β_j and the fact that only one of them can be positive implies that $\alpha_j = 0$ and $\beta_j = c_j^{\pi} = |c_j^{\pi}|$. Further, $d_j = c_j + \alpha_j - \beta_j = c_j - |c_j^{\pi}|$.

Case 2. $c_j^{\pi} < 0$. In this case, $\beta_j = 0$ and $\alpha_j = -c_j^{\pi} = |c_j^{\pi}|$. Further, $d_j = c_j + \alpha_j - \beta_j = c_j + |c_j^{\pi}|$.

Case 3. $c_j^{\pi} = 0$. In this case, $\alpha_j = \beta_j = 0$, and $d_j = c_j$.

This case analysis allows us to reformulate (4.1) as

$$\text{Maximize } -\sum_{j \in J} |c_j^{\pi}|, \tag{4.3a}$$

subject to

$$\pi_i \geq 0 \quad \text{for all } i \in B, \tag{4.3b}$$

where $c_j^{\pi} = c_j - \sum_{i \in B} a_{ij} \pi_i$. In other words, the inverse problem is to find $\pi_i \geq 0$ for all $i \in B$ so that the sum of the magnitudes of the reduced costs is minimum. We refer to the formulation (4.1) or (4.3) as the *primal inverse problem*. The preceding case analysis also implies that if π denotes the optimal solution of (4.1) or (4.3), then the optimal cost vector d^* is given by

$$d_j^* = \begin{cases} c_j - |c_j^\pi| & \text{if } c_j^\pi > 0, \\ c_j + |c_j^\pi| & \text{if } c_j^\pi < 0, \\ c_j & \text{if } c_j^\pi = 0. \end{cases} \qquad (4.4)$$

Notice that if some $c_j^\pi > 0$, then we lower the cost of the variable x_j by $|c_j^\pi|$ units, which makes its modified reduced cost zero. Similarly, if some $c_j^\pi < 0$, then we increase the cost of the variable x_j by $|c_j^\pi|$ units, which again makes its modified reduced cost zero. Finally, if $c_j^\pi = 0$, then we do not change the cost of the variable x_j since its reduced cost is already zero. Hence, when costs of the variables are modified using (4.4), the modified reduced costs of all variables become zero. These observations allow us to pose the inverse problem as: find $\pi \geq 0$ so that the modified reduced costs of all variables become zero and the sum of the modifications in reduced costs is minimum. We can summarize the preceding discussion in the form of the following theorem:

Theorem 1. *Let* **LP** *= minimize $\sum_{j \in J} c_j x_j$, subject to $\sum_{j \in J} a_{ij} x_j \geq b_i$ for all $i \in I$. Let x^0 be a feasible solution of* **LP** *and $B \subseteq I$ denote the index set of constraints that are binding with respect to x^0. Then the primal inverse problem under the L_1 norm is to find $\pi_i \geq 0$ for all $i \in B$ such that $\sum_{j \in J} |c_j^\pi|$ is minimum, where $c_j^\pi = c_j - \sum_{i \in B} a_{ij} \pi_i$. The optimal cost vector d^* is given by (4.4).*

In the formulation (4.3) we assumed that all constraints in (3.1) are of the form "$\geq$" and this lead to the non-negativity restrictions on the variables π. In case, (3.1b) has constraint i in "$\leq$" form, then the corresponding variable π_i will be non-positive; and if (3.1b) has some constraint in "=" form, then the corresponding variable π_i will be unrestricted.

We have shown that we can solve the inverse problem by solving (4.1) or (4.3), and the optimal values of π can be used to obtain the optimal cost vector using (4.4).

Instead of solving (4.1) or (4.3), we can alternatively solve the dual of (4.1) which turns out to be a variation of the original problem (3.1). The dual variables of the dual of (4.1) will be the primal variables π and they can be used to define the optimal cost vector d^*. We associate the variable y_j with the j^{th} constraint in (4.1) and then take its dual. We get the following linear programming problem:

$$\text{Minimize } \textstyle\sum_{j\in J} c_j y_j, \tag{4.5a}$$

subject to

$$\textstyle\sum_{j\in J} a_{ij}\, y_j \geq 0, \quad \text{for all } i \in B, \tag{4.5b}$$

$$-1 \leq y_j \leq 1, \quad \text{for all } j \in J. \tag{4.5c}$$

Observe that the condition (4.5c) can be restated as follows:

$$|y_j| \leq 1, \quad \text{for all } j \in J. \tag{4.5c$'$}$$

We can formulate the inverse linear programming problem in an alternate manner that may be more convenient to work with compared to the formulation in (4.5). Substituting $y_j = x_j - x_j^0$ for each $j \in J$ in (4.5) gives us the following equivalent formulation that is more similar to the original formulation of **LP**:

$$\text{Minimize } \textstyle\sum_{j\in J} c_j x_j \ - \ \sum_{j\in J} c_j\, x_j^0 \tag{4.6a}$$

subject to

$$\textstyle\sum_{j\in J} a_{ij}\, x_j \ - \ \sum_{j\in J} a_{ij}\, x_j^0 \geq 0, \quad \text{for all } i \in B, \tag{4.6b}$$

$$-1 \leq x_j - x_j^0 \leq 1, \quad \text{for all } j \in J. \tag{4.6c}$$

Using the facts that $\sum_{j\in J} c_j\, x_j^0$ is a constant, and that for each $i \in B$, $\sum_{j\in J} a_{ij}\, x_j^0 = b_i$, (4.5) can be restated as

$$\text{Minimize } \textstyle\sum_{j\in J} c_j x_j \tag{4.7a}$$

subject to

$$\textstyle\sum_{j\in J} a_{ij}\, x_j \geq b_i, \quad \text{for all } i \in B, \tag{4.7b}$$

$$x_j^0 - 1 \le x_j \le x_j^0 + 1, \quad \text{for all } j \in J. \tag{4.7c}$$

The formulations (4.5) and (4.7) of the dual of the inverse linear programming problem are equivalent to one-another. The two formulations have different primal optimal solutions and are related using the formula $x = x^0 - y$. But they have the same optimal dual solution π from which we may determine the optimal cost vector d^*.

We refer to the formulations (4.5) and (4.7) as the dual inverse problems since they are the dual of the inverse linear programming problem (4.1). We refer to the formulation (4.5) as *the 0-centered dual inverse problem*, and to the formulation (4.7) as *the x^0-centered dual inverse problem*. In the remaining discussion in this section, we shall assume that the dual inverse problem is x^0-centered but the results will also apply to 0-centered dual inverse problems in a straightforward manner.

In the formulation of the problem **LP**, we have assumed that all inequalities are of the form "≥". In case we have some "≤" inequalities, such as $\sum_{j\in J} a_{ij} x_j \le b_i$, for some $i \in I$, then we can transform it to $-\sum_{j\in J} a_{ij} x_j \ge -b_i$. If this constraint is a binding constraint with respect to the solution x^0, then the x^0-centered dual inverse problem will have the constraint $-\sum_{j\in J} a_{ij}x_j \ge -b_i$, or equivalently, $\sum_{j\in J} a_{ij} x_j \le b_i$ will be present in the formulation of the x^0-centered dual inverse problem. Hence, we need not transform a "≤" inequality in our original formulation; the dual inverse problem will have the same inequality if it is a binding constraint. Now consider the case of an equality constraint in **LP**, such as, $\sum_{j\in J} a_{ij} x_j = b_i$. We may replace this constraint by the two constraints, $\sum_{j\in J} a_{ij} x_j \ge b_i$ and $-\sum_{j\in J} a_{ij} x_j \ge -b_i$. Since x^0 is a feasible solution of **LP** (that is, satisfies $\sum_{j\in J} a_{ij} x_j = b_i$), both the preceding inequalities will be binding, which is equivalent to the constraint $\sum_{j\in J} a_{ij} x_j = b_i$ in the dual inverse problem. Consequently, if we have an equation in **LP**, the same equation will always be present in the x^0-centered dual inverse problem. We summarize our discussion in this section in the form of the following theorem:

Theorem 2. *Let* LP = *minimize* $\sum_{j\in J} c_j x_j$, *subject to* $\sum_{j\in J} a_{ij} x_j \begin{Bmatrix} \geq \\ = \\ \leq \end{Bmatrix} b_i$ *for all* $i \in I$. *Let* x^0 *be a feasible solution of* LP *and* $B \subseteq I$ *denote the index set of constraints that are binding with respect to* x^0. *Then the inverse linear programming problem under the* L_1 *norm is the dual of the following problems:*

0-centered dual inverse problem: *minimize* $\sum_{j\in J} c_j y_j$, *subject to* $\sum_{j\in J} a_{ij} y_j \begin{Bmatrix} \geq \\ = \\ \leq \end{Bmatrix} 0$ *for all* $i \in B$, *and* $-1 \leq y_j \leq 1$ *for all* $j \in J$.

x^0-centered dual inverse problem: *minimize* $\sum_{j\in J} c_j x_j$, *subject to* $\sum_{j\in J} a_{ij} x_j \begin{Bmatrix} \geq \\ = \\ \leq \end{Bmatrix} b_i$ *for all* $i \in B$, *and* $x_j^0 - 1 \leq x_j \leq x_j^0 + 1$ *for all* $j \in J$.

Let π *denote the optimal dual variables associated with the binding constraints. Then the optimal cost vector* d^* *is given by (4.4).*

Weighted Inverse Problems

We now consider the inverse linear programming problem under the weighted L_1 norm, that is, where the objective function is $\sum_{j\in J} w_j |d_j - c_j|$, with w_j's being specified constants. We can use the same approach as for the unit weight case to formulate inverse problems. The primal inverse problem for the weighted case will be the same as (4.3) except that the objective function (4.3a) is replaced by the following objective: Minimize $\sum_{j\in J} w_j |c_j^{\pi}|$. The 0-centered dual inverse problem for the weighted case will be the same as (4.5) except that (4.5c) is replaced by $-w_j \leq y_j \leq w_j$, and the x^0-centered dual inverse problem for the weighted case will be the same as (4.7) except that (4.7c) is replaced by $x_j^0 - w_j \leq x_j \leq x_j^0 + w_j$.

5. Inverse Bounded Variable Linear Programming Problems under L_1 Norm

In the linear programming problem (3.1) studied by us, each constraint was an inequality constraint but we did not have any additional lower or upper bound restrictions on variables. Rather, we assumed that any such restrictions would be treated as regular constraints. We will now drop this assumption and will consider the lower and upper bound restrictions on variables explicitly. We will consider the following linear programming problem:

$$\text{Minimize } \textstyle\sum_{j\in J} c_j x_j, \tag{5.1a}$$

subject to

$$\textstyle\sum_{j\in J} a_{ij}\, x_j \geq b_i, \qquad \text{for all } i \in I, \tag{5.1b}$$

$$x_j \geq 0, \qquad \text{for all } j \in L, \tag{5.1c}$$

$$-x_j \geq -u_j, \qquad \text{for all } j \in U, \tag{5.1d}$$

where 0 and u_j, respectively, denote lower and upper bounds on the values of the variable x_j. We refer to (5.1) as the bounded variable linear programming problem. Let x^0 be a given feasible solution of (5.1) which we wish to make optimal by perturbing the cost coefficients of the variables. Let B denote the index set of binding constraints in (5.1b) with respect to x^0, L denote the index set of binding constraints in (5.1c) (that is, $L = \{j \in J: x_j^0 = 0\}$), and U denote the index set of binding constraints in (5.1) (that is, $U = \{j \in J: x_j^0 = u_j\}$). Let $F = \{j \in J: 0 < x_j^0 < u_j\}$. Notice that the sets F, L, and U are mutually exclusive and exhaustive. As earlier, we associate the dual variable $\pi_i \geq 0$ for the i^{th} constraint in (5.1b). Further, we associate the dual variable $\lambda_j \geq 0$ for the j^{th} constraint in (5.1c) and $\varphi_j \geq 0$ with the j^{th} constraint in (5.1d). In this case, it follows from discussion in Section 3 that the inverse of (5.1) is the following linear program:

$$\text{Maximize } -\textstyle\sum_{j\in J} \alpha_j - \sum_{j\in J} \beta_j,$$

or, equivalently,

$$\text{minimize } \textstyle\sum_{j\in J} \alpha_j + \sum_{j\in J} \beta_j, \tag{5.2a}$$

subject to

$$\textstyle\sum_{i\in B} a_{ij}\, \pi_i - \alpha_j + \beta_j + \lambda_j = c_j, \text{ for all } j \in L, \tag{5.2b}$$

$$\textstyle\sum_{i\in B} a_{ij}\, \pi_i - \alpha_j + \beta_j - \varphi_j = c_j, \text{ for all } j \in U, \tag{5.2c}$$

$$\textstyle\sum_{i\in B} a_{ij}\, \pi_i - \alpha_j + \beta_j \qquad = c_j, \text{ for all } j \in F, \tag{5.2d}$$

$$\pi_i \geq 0 \text{ for all } i \in B;\ \alpha_j \geq 0 \text{ and } \beta_j \geq 0 \text{ for all } j \in J, \tag{5.2e}$$

$$\lambda_j \geq 0 \text{ for all } j \in L; \text{ and } \varphi_j \geq 0 \text{ for all } j \in U. \tag{5.2f}$$

We will now simplify (5.2). As earlier, both of α_j and β_j cannot take positive values. We can restate (5.2b), (5.2c) and (5.2d) as

$$-\alpha_j + \beta_j = c_j^{\pi} - \lambda_j, \qquad \text{for all } j \in L, \tag{5.3a}$$

$$-\alpha_j + \beta_j = c_j^{\pi} + \varphi_j, \qquad \text{for all } j \in U, \tag{5.3b}$$

$$-\alpha_j + \beta_j = c_j^{\pi}, \qquad \text{for all } j \in F, \tag{5.3c}$$

where $c_j^{\pi} = c_j - \Sigma_{i\in B}\, a_{ij}\, \pi_i$. There are three cases to consider.

Case 1. $c_j^{\pi} > 0$. The non-negativity of α_j and β_j and the fact that we wish to minimize $\alpha_j + \beta_j$ implies that (i) if $j \in L$ then $\lambda_j = c_j^{\pi} = |c_j^{\pi}|$, $\alpha_j = \beta_j = 0$ and $d_j = c_j$; and (ii) if $j \in F \cup U$ then $\alpha_j = \varphi_j = 0$, and $\beta_j = c_j^{\pi} = |c_j^{\pi}|$. Further, $d_j = c_j - |c_j^{\pi}|$.

Case 2. $c_j^{\pi} < 0$. In this case, (i) if $j \in U$ then $\varphi_j = -c_j^{\pi} = |c_j^{\pi}|$, $\alpha_j = \beta_j = 0$ and $d_j = c_j$; and (ii) if $j \in F \cup L$ then $\beta_j = \lambda_j = 0$ and $\alpha_j = -c_j^{\pi} = |c_j^{\pi}|$. Further, $d_j = c_j + |c_j^{\pi}|$.

Case 3. $c_j^{\pi} = 0$. In this case, $\alpha_j = \beta_j = \lambda_j = \varphi_j = 0$, and $d_j = c_j$.

This case analysis allows us to reformulate (5.2) as

$$\text{Minimize } \Sigma_{j\in L} \max\{0, -c_j^{\pi}\} + \Sigma_{j\in F} |c_j^{\pi}| + \Sigma_{j\in U} \max\{0, c_j^{\pi}\}, \tag{5.4a}$$

subject to

$$\pi_i \geq 0 \quad \text{for all } i \in B, \tag{5.4b}$$

where $c_j^{\pi} = c_j - \Sigma_{i\in B}\, a_{ij}\, \pi_i$. The preceding case analysis also implies that if π denotes the optimal solution of (5.2), then the optimal cost vector d^* is given by

$$d^*_j = \begin{cases} c_j - |c^\pi_j| & \text{if } c^\pi_j > 0 \text{ and } x^0_j > 0, \\ c_j + |c^\pi_j| & \text{if } c^\pi_j < 0 \text{ and } x^0_j < u_{ij}, \\ c_j & \text{otherwise.} \end{cases} \tag{5.5}$$

We next obtain the formulation of the x^0-centered dual inverse linear programming problem for the bounded variable case. Notice that for each $j \in L$, $x_j \geq 0$ is a binding constraint, and for each $j \in U$, $x_j \leq u_j$ is a binding constraint. Applying Theorem 2 gives the following linear programming problem:

$$\text{Minimize } \textstyle\sum_{j \in J} c_j x_j, \tag{5.6a}$$

subject to

$$\textstyle\sum_{j \in J} a_{ij}\, x_j \geq b_i, \qquad \text{for all } i \in B, \tag{5.6b}$$

$$0 \leq x_j \leq 1, \qquad \text{for all } j \in L, \tag{5.6c}$$

$$u_j - 1 \leq x_j \leq u_j, \qquad \text{for all } j \in U, \tag{5.6d}$$

$$x^0_j - 1 \leq x_j \leq x^0_j + 1, \quad \text{for all } j \in F. \tag{5.6e}$$

0-1 Linear Programming Problem

We will now consider a special case of the bounded variable linear programming problem where each upper bound equals one, and there always exists an integer optimal solution. We refer to such a linear programming problem as *a 0-1 linear programming problem*. Several combinatorial optimization problems, such as, the single-source single-sink shortest path problem, the assignment problem, and the minimum cut problem, can be formulated as 0-1 linear programming problems. Let x^0 be a 0-1 feasible solution of a 0-1 linear programming problem which we wish to make optimal by perturbing the cost vector c to d. Let B denote the index set of constraints binding with respect to the solution x^0. Since x^0 is a 0-1 solution, each index $j \in J$ either belongs to L or U, and in both the cases, (5.6c) or case (5.6d), reduces to $0 \leq x_j \leq 1$. We thus get the following x^0-centered dual inverse problem:

$$\text{Minimize } \textstyle\sum_{j \in J} c_j x_j \tag{5.7a}$$

subject to

$$\sum_{j \in J} a_{ij} x_j \geq b_i, \qquad \text{for all } i \in B, \tag{5.7b}$$

$$0 \leq x_j \leq 1, \qquad \text{for all } j \in J, \tag{5.7c}$$

which is the same as the original problem except that the non-binding constraints with respect to x^0 have been eliminated. In the additional case when all constraints are binding (for example, when each constraint in (5.7b) is an equality constraint), B = I and its x^0-centered dual inverse problem is the same as the original problem. In the case of the 0-1 linear programming problem, we can restate the expression for computing the optimal cost vector d^*. Let x^* be an optimal solution of (5.7) and π denote the optimal dual variables associated with the constraints in (5.7b). It follows from the linear programming theory that (i) $c_j^\pi < 0$ if and only if $x_j^* = u_j$, and (ii) $c_j^\pi > 0$ if and only if $x_j^* = 0$. Using these results in (5.5) yields the following optimal cost vector:

$$d_j^* = \begin{cases} c_j - |c_j^\pi| & \text{for all j satisfying } x_j^0 = 1 \text{ and } x_j^* = 0, \\ c_j + |c_j^\pi| & \text{for all j satisfying } x_j^0 = 0 \text{ and } x_j^* = 1, \\ c_j & \text{for all j satisfying } x_j^0 = x_j^*. \end{cases} \tag{5.8}$$

6. SOLVING THE INVERSE LINEAR PROGRAMMING PROBLEM UNDER THE L_∞ NORM

In this section, we study the inverse of the linear programming problem **LP** under the L_∞ norm, called the *minimax inverse linear programming problem*. In this problem, we wish to obtain an inverse feasible cost vector d that minimizes $\max\{|d_j - c_j| : j \in J\}$. It follows from (3.9) that the minimax inverse linear programming problem can be formulated as the following mathematical programming problem:

$$\text{Minimize} \max_{j \in J} \{|d_j - c_j|\} \tag{6.1a}$$

subject to

$$\sum_{i \in B} a_{ij} \pi_i = d_j, \quad \text{for all } j \in J, \tag{6.1b}$$

$$\pi_i \geq 0, \quad \text{for all } i \in B. \tag{6.1c}$$

This mathematical program is not a linear program since it contains absolute signs on terms in the objective function and a maximization of terms instead of summation of terms; however, it can be converted to a linear programming problem by using well known transformations. To eliminate the absolute signs in the objective function, we replace $|d_j - c_j|$ by $\alpha_j + \beta_j$, subject to $d_j - c_j = \alpha_j - \beta_j$, $\alpha_j \geq 0$ and $\beta_j \geq 0$. Further, to eliminate the maximization of the terms, we introduce a variable θ and add the constraints $\alpha_j + \beta_j \leq \theta$ for each $j \in J$ to ensure that each term is less than or equal to θ. We also convert the minimization form of the objective function into the maximization form. This gives us the following linear programming problem:

$$\text{Maximize } -\theta \tag{6.2a}$$

subject to

$$\sum_{i \in B} a_{ij}\, \pi_i - \alpha_j + \beta_j = c_j, \qquad \text{for all } j \in J, \tag{6.2b}$$

$$\alpha_j + \beta_j - \theta \leq 0, \qquad \text{for all } j \in J, \tag{6.2c}$$

$$\pi_i \geq 0 \text{ for all } i \in B;\ \alpha_j \geq 0 \text{ and } \beta_j \geq 0 \text{ for all } j \in J. \tag{6.2d}$$

We will now simplify (6.2). We first note that there exists an optimal solution in which for every j both of α_j and β_j cannot take positive values, since otherwise we can reduce one of them to zero without violating any constraint and without worsening the objective function value. We can restate (6.2) as

$$-\alpha_j + \beta_j = c_j^{\pi}, \text{ for all } j \in J, \tag{6.3}$$

where $c_j^{\pi} = c_j - \sum_{i \in B} a_{ij}\, \pi_i$. There are three cases to consider.

Case 1. $c_j^{\pi} > 0$. The non-negativity of α_j and β_j and the fact that only one of them can be positive imply that $\alpha_j = 0$ and $\beta_j = c_j^{\pi} = |c_j^{\pi}|$. In this case, the constraint (6.2c) becomes $c_j^{\pi} \leq \theta$. Further, $d_j = c_j + \alpha_j - \beta_j = c_j - |c_j^{\pi}|$.

Case 2. $c_j^{\pi} < 0$. In this case, $\beta_j = 0$ and $\alpha_j = -c_j^{\pi} = |c_j^{\pi}|$. In this case, the constraint (6.2c) becomes $c_j^{\pi} \geq -\theta$. Further, $d_j = c_j + \alpha_j - \beta_j = c_j + |c_j^{\pi}|$.

Case 3. $c_j^\pi = 0$. In this case, $\alpha_j = \beta_j = 0$, and $d_j = c_j$. In this case, the constraint (6.2c) is satisfied.

The preceding analysis allows us to formulate (6.2) as the following linear program:

$$\text{Maximize } -\theta, \tag{6.4a}$$

subject to

$$-\theta \le c_j^\pi \le \theta \quad \text{for all } j \in J. \tag{6.4b}$$

or, alternatively,

$$|c_j^\pi| \le \theta, \qquad \text{for all } j \in J. \tag{6.4b'}$$

In other words, the minimax inverse problem reduces to finding the smallest value of θ such that the largest magnitude of any reduced cost is at most θ. We refer to the formulation (6.4) as the primal minimax inverse problem. If the optimal objective function of (6.4) is zero, then it implies that x^0 is an optimal solution of (3.1) and hence $d^* = c$. If not, then costs must be changed. The previous case analysis also implies that the optimal cost vector $d_j = c_j + \alpha_j - \beta_j$ is given by

$$d_j^* = \begin{cases} c_j - |c_j^\pi| & \text{if } c_j^\pi > 0, \\ c_j + |c_j^\pi| & \text{if } c_j^\pi < 0, \\ c_j & \text{if } c_j^\pi = 0, \end{cases} \tag{6.5}$$

which is the same as in the case of L_1 norm; however the optimal value of π will typically be different. As in the case of the L_1 norm, the cost coefficients are modified so that the reduced cost of each variable becomes zero. We can summarize the preceding discussion in the form of the following theorem:

Theorem 4. *Let LP = minimize $\sum_{j\in J} c_j x_j$, subject to $\sum_{j\in J} a_{ij} x_j \ge b_i$ for all $i \in I$. Let x^0 be a feasible solution of LP and $B \subseteq I$ denote the index set of constraints that are binding with respect to x^0. Then the primal minimax inverse linear programming problem under* *the L_∞ norm is to find $\pi_i \geq 0$ for all $i \in B$ such that $\max\{|c_j^\pi| : j \in J\}$ is minimum, where $c_j^\pi = c_j - \Sigma_{i\in B} a_{ij}\,\pi_i$. The optimal cost vector d^* is given by (6.5).*

By taking the dual of (6.4), we can obtain an equivalent formulation of the minimax inverse problem. We associate the variable y_j^+ with the constraint $-\theta \leq c_j^\pi = c_j - \Sigma_{i\in B} a_{ij}\,\pi_i$, and the variable y_j^- with the constraint $c_j^\pi = c_j - \Sigma_{i\in B} a_{ij}\,\pi_i\, c_j \leq \theta$. The dual of (6.4) is the following linear programming problem:

$$\text{Minimize } \Sigma_{j\in J}\, c_j\,(y_j^+ - y_j^-), \tag{6.6a}$$

subject to

$$\Sigma_{j\in J}\, a_{ij}\,(y_j^+ - y_j^-) \geq 0, \quad \text{for all } i \in B, \tag{6.6b}$$

$$\Sigma_{j\in J}\,(y_j^+ + y_j^-) \leq 1. \tag{6.6c}$$

By letting $y_j = (y_j^+ + y_j^-)$, we can reformulate (6.6) as follows:

$$\text{Minimize } \Sigma_{j\in J}\, c_j y_j, \tag{6.7a}$$

subject to

$$\Sigma_{j\in J}\, a_{ij}\, y_j \geq 0, \quad \text{for all } i \in B, \tag{6.7b}$$

$$\Sigma_{j\in J}\, |y_j| \leq 1. \tag{6.7c}$$

If π denotes the optimal dual variables associated with (6.6b) or (6.7b), then the optimal cost vector d^* can be computed using (6.5). Notice that the formulation (6.7) has a close resemblance with the formulation for the inverse problem under the L_1 norm: in place of the constraints $|y_j| \leq 1$ for all $j \in J$, we have just one constraint $\Sigma_{j\in J}\, |y_j| \leq 1$. We refer to the formulation (6.7) as *the 0-centered minimax dual inverse problem*. We can obtain the *x^0-centered minimax dual inverse problem* by substituting $y_j = x_j - x_j^0$ for each $j \in J$ in (6.7). This gives us the following equivalent formulation of the minimax inverse problem:

$$\text{Minimize } \sum_{j\in J} c_j x_j \tag{6.8a}$$

subject to

$$\sum_{j\in J} a_{ij}\, x_j \geq b_i, \qquad \text{for all } i \in B, \tag{6.8b}$$

$$\sum_{j\in J} |x_j - x_j^0| \leq 1. \tag{6.8c}$$

We have assumed so far in this section that all the constraints in (3.1) are of the "≥" form. A similar analysis will work for the "≤" or for the "=" type of constraints. We summarize our discussion in the form of the following theorem:

Theorem 4. *Let* ***LP*** *= minimize $\sum_{j\in J} c_j x_j$, subject to $\sum_{j\in J} a_{ij}\, x_j \begin{Bmatrix} \geq \\ = \\ \leq \end{Bmatrix} b_i$ for all $i \in I$. Let x^0 be a feasible solution of* ***LP*** *and $B \subseteq I$ denote the index set of constraints that are binding with respect to x^0. Then the inverse problem under the L_∞ norm is the dual of the following problems:*

0-centered minimax dual inverse problem: *minimize $\sum_{j\in J} c_j y_j$, subject to $\sum_{j\in J} a_{ij}\, y_j \begin{Bmatrix} \geq \\ = \\ \leq \end{Bmatrix} 0$ for all $i \in B$, and $\sum_{j\in J} |y_j| \leq 1$.*

x^0-centered minimax dual inverse problem: *minimize $\sum_{j\in J} c_j x_j$, subject to $\sum_{j\in J} a_{ij}\, x_j \begin{Bmatrix} \geq \\ = \\ \leq \end{Bmatrix} b_i$ for all $i \in B$, and $\sum_{j\in J} |x_j - x_j^0| \leq 1$.*

Let π denote the optimal dual variables associated with the binding constraints. Then the optimal cost vector d^* is given by (6.5).

Bounded Variable Linear Programming Problem

We next consider the minimax inverse versions of the bounded variable linear programming problem (5.1). We can consider this case by specializing Theorem 4, as we did in Section 5. We will omit the details and state only the final result. It can be shown that the primal minimax inverse problem in the bounded variable is the following linear programming problem:

$$\text{Maximize } -\theta, \tag{6.9a}$$

subject to

$$c_j^\pi \le \theta \qquad \text{for all } j \in J\backslash L. \tag{6.9b}$$

$$-\theta \le c_j^\pi \qquad \text{for all } j \in J\backslash U, \tag{6.9b}$$

and the optimal cost vector d^* is given by

$$d_j^* = \begin{cases} c_j - |c_j^\pi| & \text{if } c_j^\pi > 0 \text{ and } x_j^0 > 0, \\ c_j + |c_j^\pi| & \text{if } c_j^\pi < 0 \text{ and } x_j^0 < u_{ij}, \\ c_j & \text{otherwise.} \end{cases} \tag{6.10}$$

which is the same as in the case of the L_1 norm. As before, one would expect the optimal value of π to be different.

Weighted Minimax Inverse Linear Programming Problem

In our preceding analysis, we have considered the unit weight minimax inverse linear programming problem. In the weighted minimax inverse linear programming problem, the objective function is to minimize $\max\{w_j|d_j - c_j| : j \in J\}$, where $w_j \ge 0$ for each $j \in J$. The formulations of the weighted minimax inverse linear programming problem are exactly the same as the formulations for the unit weight case except the following changes: (i) in the formulation (6.4), the constraint $-\theta \le c_j^\pi \le \theta$ is replaced by the constraint $-\theta \le w_j c_j^\pi \le \theta$; (ii) in the formulation (6.7), the $\sum_{j\in J} |y_j| \le 1$ is replaced by the constraint $\sum_{\{j\in J: w_j \ne 0\}} |y_j|/w_j \le 1$ and $y_j = 0$ if $w_j = 0$; and (iii) in the formulation (6.8), the constraint $\sum_{j\in J} |x_j - x_j^0| \le 1$ is replaced by the constraint $\sum_{\{j\in J: w_j \ne 0\}} |x_j - x_j^0|/w_j \le 1$, and $x_j = x_j^0$ if $w_j = 0$.

7. The General Inverse Optimization Problem

In this section, we consider the general inverse optimization problem and show that (under reasonable regularity conditions) that if the problem **P** is polynomially solvable, then its inverse versions under L_1 and L_∞ norms are also polynomially solvable. This result makes use of the ellipsoid algorithm, and we refer the reader to the books by Schrijver [1986] and Grotschel, Lovasz, and Schrijver [1986].

Recall from Section 1 that we denote by **S** the set of feasible solutions, and **P** = $\min\{cx : x \in \mathbf{S}\}$. We denote by Q^n, the set of all rational numbers in the n dimensional space. Suppose that a polyhedron $\mathbf{D} \subseteq Q^n$ is defined by rational linear inequalities in terms of the rationals of size at most φ. On the polyhedron **D**, the *separation problem*, and *optimization problem*, can be defined as follows.

Separation Problem: *Given a polyhedron* $\mathbf{D} \subseteq R^n$ *and a vector* $d' \in R^n$, *the separation problem is to:*

(a) either decide that $d' \in \mathbf{D}$; *or* *(7.1a)*

(b) find a vector $y \in Q^n$ *such that* $dy < d'y$ *for all* $d \in \mathbf{D}$. *(7.1b)*

Optimization Problem: *Given a polyhedron* $\mathbf{D} \subseteq R^n$ *and a vector* $r \in Q^n$, *conclude with one of the following:*

(a) *give a vector* $d^* \in \mathbf{D}$ *with* $rd^* = \min\{rd : d \in \mathbf{D}\}$;
there exists a vector $d \in \mathbf{D}$ *with unbounded objective function value;*
(b) *assert that* **D** *is empty;*

It is well known in the case that **D** is specified by a set of linear constraints, then to solve the separation problem it is sufficient to check whether the given solution d' satisfies all the constraints. If yes, then $d' \in \mathbf{D}$, satisfying (7.1a); otherwise, a violated constraint gives a "separator vector" y satisfying (7.1b). We also use the following well known result:

Theorem 5 (Grotschel, Lovasz, and Schrijver [1986]). *The optimization problem can be solved in time polynomially bounded by n, φ, the size of c, and the running time of the separation problem.*

We will show that for inverse **P** under the L_1 or L_∞ norms, the separation problem reduces to solving a single instance of **P**. Therefore, if **P** can be solved in polynomial time, Theorem 5 implies that inverse **P** under the L_1 or L_∞ norms can also be solved in polynomial time. We will first consider inverse **P** under the L_1 norm, which can be formulated as:

$$\text{Minimize } \sum_{j \in J} |d_j - c_j|, \tag{7.2a}$$

subject to

$$dx^0 \le dx \qquad \text{for all } x \in \mathbf{S}. \tag{7.2b}$$

The mathematical program (7.2) is not a linear programming problem, but can be transformed to one by introducing some additional variables and constraints. The resulting linear programming problem is to

$$\text{Minimize } \sum_{j \in J} z_j \tag{7.3a}$$

subject to

$$dx^0 \le dx \qquad \text{for all } x \in \mathbf{S}, \tag{7.3b}$$

$$d_j - c_j \le z_j \qquad \text{for all } j \in J, \tag{7.3c}$$

$$c_j - d_j \le z_j \qquad \text{for all } j \in J. \tag{7.3d}$$

It is easy to see that (7.3) is equivalent to (7.2). The constraints in (7.3c) and (7.3d) imply that $z_j \ge |d_j - c_j|$. Further, since we minimize $\sum_{j \in J} z_j$, each z_j will equal $|d_j - c_j|$ in an optimum solution. We will show how can we solve the separation problem for the set of constraints in (7.3) in polynomial time. Let $\overline{\mathbf{D}}$ denote the polyhedron defined by the feasible solutions of (7.3). We assume that all data in (7.3) is rational and the largest number in the data is φ. Given a proposed solution $(d', z') \in$, we can easily check in linear time whether the solution (d', z') belongs to $\overline{\mathbf{D}}$ by checking whether it satisfies (7.3c) and (7.3d). To check whether the solution d' satisfies (7.3b), we solve the problem **P** with d' as the cost vector. Let x' denote the resulting optimal solution. If $d'x^0 = d'x'$, then d' satisfies (7.3); otherwise we have found a violated inequality $d'x^0 > d'x'$. Thus we can solve the separation problem for (7.3) by solving a single instance of **P**. This result in view of Theorem 5 implies that inverse **P** under the L_1 norm is polynomially solvable.

We now consider inverse **P** under the L_∞ norm. In this case, we wish to minimize $\max\{|d_j - c_j|: j \in J\}$, subject to $dx^0 \le dx$, for all $x \in$ **S**. This mathematical program can be formulated as the following linear programming problem:

$$\text{Minimize } z \tag{7.4a}$$

subject to

$$dx^0 \le dx \quad \text{for all } x \in \mathbf{S}, \tag{7.4b}$$

$$d_j - c_j \le z \quad \text{for all } j \in J, \tag{7.4c}$$

$$c_j - d_j \le z \quad \text{for all } j \in J. \tag{7.4d}$$

Using the same technique as in the case of the L_1 norm, it can be shown that we can solve the separation problem for (7.4) by solving a single instance of problem P. We summarize the preceding discussion as the following theorem:

Theorem 6. *If a problem **P** is polynomially solvable for each linear cost function, then inverse **P** as well as minimax inverse **P** are polynomially solvable.*

This result establishes the polynomial solvability of large classes of inverse optimization problems; however, the ellipsoid algorithm is not yet practical for large problems. Moreover, for many specific classes of problems, such as network flow problems, one may obtain improved polynomial time algorithms.

ACKNOWLEDGEMENTS

We gratefully acknowledge the support from the Office of Naval Research under contract ONR N00014-96-1-0051 as well as a grant from the United Parcel Service. We also acknowledge the help of Don Wagner who raised some perceptive and fundamental questions that led to the pursuit of the research reported in this paper.

REFERENCES

Ahuja, R. K., T. L. Magnanti, and J. B. Orlin. 1993. *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, NJ.

Ahuja, R. K., and J. B. Orlin. 1997. Solving the convex ordered set problem with applications to isotonic regression. Working Paper, Sloan School of Management, MIT, Cambridge, MA.

Ahuja, R. K., and J. B. Orlin. 1998a. A fast algorithm for the bipartite node weighted matching problem on path graphs with application to the inverse spanning tree problem. Working Paper, Sloan School of Management, MIT, Cambridge, MA.

Ahuja, R. K., and J. B. Orlin. 1998b. Inverse optimization, Part 2: Network flow problems. Working Paper, Sloan School of Management, MIT, Cambridge, MA.

Ahuja, R. K., and J. B. Orlin. 1998c. Combinatorial algorithms for inverse network flow problems. Working Paper, Sloan School of Management, MIT, Cambridge, MA.

Barlow, R. E., D. J. Bartholomew, D. J. Bremner, and H. D. Brunk. 1972. *Statistical Inference under Order Restrictions.* Wiley, New York.

Best, M. J., and N. Chakravarti. 1990. Active set algorithms for isotonic regression: A unifying framework. *Mathematical Programming* **47**, 425-439.

Burton, D., B. Pulleyblank, and Ph. L. Toint. 1997. The inverse shortest paths problem with upper bounds on shortest paths costs. In *Network Optimization*, edited by P. Pardalos, D. W. Hearn, and W. H. Hager, *Lecture Notes in Economics and Mathematical Systems*, Volume 450, pp. 156-171.

Burton, D., and Ph. L. Toint. 1992. On an instance of the inverse shortest paths problem. *Mathematical Programming* **53**, 45-61.

Burton, D., and Ph. L. Toint. 1994. On the use of an inverse shortest paths algorithm for recovering linearly correlated costs. *Mathematical Programming* **63**, 1-22.

Cai, M., and Y. Li. 1995. Inverse matroid intersection problem. Research Report, Institute of System Science, Academia Sinica, Beijing, China. To appear *in ZOR-Mathematical Methods of Operations Research.*

Cai, M., X. Yang, and Y. Li. 1996. Inverse polymatroidal flow problem. Research Report, Institute of System Science, Academia Sinica, Beijing, China.

Cai, M. and X. Yang. 1994. Inverse shortest path problems. Technical Report, Institute of Systems Sciences, Academia Sinica, Beijing, China.

Dial, B. 1997. Minimum-revenue congestion pricing, Part 1: A fast algorithm for the single-origin case. Technical Report, The Volpe National Transportation Systems Center, Kendall Square, Cambridge, MA 02142.

Greenberg, H. J. 1997. An annotated bibliography for post-solution analysis in mixed integer programming and combinatorial optimization. *Advances in Computational and Stochastic Optimization, Logic Programming, and Heuristic Search*, Edited by D. L. Woodruff, Kluwer Academic Publishers.

Grotschel, M., L. Lovasz, and A. Schrijver. 1986. *The Ellipsoid Method and Combinatorial Optimization*. Springer, Heidelberg.

Hu, Z., and Z. Liu. 1995. A strongly polynomial algorithm for the inverse shortest arborescence problem. Working Paper, Institute of Systems science, Academia Sinica, Beijing, China.

Huang, S., and Z. Liu. 1995a. On the inverse problem of k-matching of bipartite graph. Working Paper, Department of Management, School of Business and Management, Hong Kong University of Science and Technology, Hong Kong.

Huang, S., and Z. Liu. 1995b. On the inverse version of the minimum cost flow problem. Working Paper, Dept. of ISMT, School of Business and Management, Hong Kong University of Science and Technology, Hong Kong.

Ma, Z., S. Xu, and J. Zhang. 1996. Algorithms for inverse minimum spanning tree problem, Working Paper, Department of Mathematics, City Polytechnic of Hong Kong, Hong Kong.

Neumann-Denzau, G., and J. Behrens. 1984. Inversion of seismic data using tomographical reconstruction techniques for investigations of laterally inhomogeneous media. *Geophysical Journal of the Royal Astronomical Society* **79**, 305-315.

Nolet. G. 1987. *Seismic Tomography.* Reidel, Dordrecht.

Robertson, T., F. T. Wright, and R. L. Dykstra. 1988. *Order Restricted Statistical Inference*. John Wiley & Sons, New York.

Schrijver, A. 1986. *Theory of Linear and Integer Programming*. John Wiley & Sons.

Sheffi, Y. 1985. *Urban Transportation Networks*. MIT Press, Cambridge, MA.

Sokkalingam, P.T. 1996. *The Minimum Cost Flow Problem : Primal Algorithms and Cost Perturbations*. Unpublished Dissertation, Department of Mathematics, Indian Institute of Technology, Kanpur, INDIA.

Sokkalingam, P.T., R. K. Ahuja, and J. B. Orlin. 1996. Solving the inverse spanning tree problems through network flow techniques. Working Paper, Sloan School of Management, MIT, Cambridge, MA. To appear in *Operations Research.*

Sotskov, Y. N., V. K. Leontev, and E. N. Gordeev. 1995. Some concepts of stability analysis in combinatorial optimization. *Discrete Applied Mathematics* **58**, 169-190.

Sotskov, Y. N., V. S. Tanaev, and F. Werner. 1997. Stability radius of an optimal schedule: A survey and recent developments. In *Industrial Applications of Combinatorial Optimization*, Edited by G. Yu, Kluwer Academic Press, Boston.

Tarantola, A. 1987. *Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation.* Elsevier, Amsterdam.

Woodhouse, J. H., and A. M. Dziewonski. 1984. Mapping the upper mantle: Three dimensional modeling of Earth structure by inversion of seismic waveforms. *Journal of Geophysical Research* **89** (B7), 5953-5986.

Xu, S., and J. Zhang. 1995. An inverse problem of the weighted shortest path problem. *Japanese Journal of Industrial and Applied Mathematics* **12**, 47-59.

Yang, C., and J. Zhang. 1996. Inverse maximum capacity path with upper bound contraints. To appear in *OR Spektrum*.

Yang, C., J. Zhang, and Z. Ma. 1997. Inverse maximum flow and minimum cut problem. *Optimization* **40**, 147-170.

Zhang, J., Z. Ma, and C. Yang. 1995. A column generation method for inverse shortest path problems, *ZOR-Mathematical Methods for Operations Research* **41**, 347-358.

Zhang, J., and and M. C. Cai. 1998. Inverse problem of minimum cuts. *Mathematical Methods of Operations Research* **47**, No. 1.