Quantile Inverse Optimization: Improving Stability in Inverse Linear Programming

Zahed Shahmoradi, Taewoo Lee

Please scroll down for article—it is on subsequent pages

With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes.
For more information on INFORMS, its publications, membership, or meetings visit http://www.informs.org

http://pubsonline.informs.org/journal/opre

Methods

Quantile Inverse Optimization: Improving Stability in Inverse Linear Programming

Zahed Shahmoradi,[a] Taewoo Lee[a]

[a] Industrial Engineering, University of Houston, Houston, Texas 77204
Contact: zshahmoradi@uh.edu, https://orcid.org/0000-0003-1419-681X (ZS); tlee6@uh.edu, https://orcid.org/0000-0003-2222-8201 (TL)

Abstract. Inverse linear programming (LP) has received increasing attention because of its potential to infer efficient optimization formulations that can closely replicate the behavior of a complex system. However, inversely inferred parameters and corresponding forward solutions from the existing inverse LP methods can be highly sensitive to noise, errors, and uncertainty in the input data, limiting their applicability in data-driven settings. We introduce the notion of inverse and forward stability in inverse LP and propose a novel inverse LP method that determines a set of objective functions that are stable under data imperfection and generate forward solutions close to the relevant subset of the data. We formulate the inverse model as a large-scale mixed-integer program (MIP) and elucidate its connection to biclique problems, which we exploit to develop efficient algorithms that solve much smaller MIPs instead to construct a solution to the original problem. We numerically evaluate the stability of the proposed method and demonstrate its use in the diet recommendation and transshipment applications.

Keywords: inverse optimization • inverse linear programming • online learning

1. Introduction

Given a set of observed decisions as input data, inverse optimization infers parameters of a "forward" optimization problem, for example, objective function coefficients, that make the given decisions optimal or near optimal. By doing so, inverse optimization allows the forward problem to capture the preferences or utilities of the decision maker (DM) and reproduce the decisions accordingly. Inverse optimization has recently found numerous applications, including finance (Bertsimas et al. 2012), transportation (Bertsimas et al. 2015), the electricity market (Birge et al. 2017, Saez-Gallego and Morales 2017), incentive design (Aswani et al. 2019), and healthcare (Erkin et al. 2010, Lee et al. 2013).

Various inverse optimization models have been developed for different types of forward problems, including network optimization (e.g., Ahuja and Orlin 2001, Heuberger 2004), linear programs (Ahuja and Orlin 2001, Troutt et al. 2008, Chan et al. 2014, Ghate 2015), conic programs (Iyengar and Kang 2005), convex programs (Keshavarz et al. 2011), integer and mixed-integer programs (MIPs) (Schaefer 2009, Wang 2009, Lamperski and Schaefer 2015), and multicriteria optimization (Chan and Lee 2018, Naghavi et al. 2019). The underlying assumption of these studies is that the choice of the inverse model depends on which type of the forward problem is to be used for reproducing the decisions; for example, if the user wants to derive a linear programming (LP) formulation that can replace a complex decision model, inverse LP is used to fit a linear objective function to the observed decisions.

As the goal of inverse optimization is to fit a model to given data, recent studies naturally reveal and leverage the connection between inverse optimization and regression. Aswani et al. (2018) highlight that the residuals used in their inverse model are similar to those in regression and propose a loss function analogous to the sum of squared errors in ordinary least squares. Bertsimas et al. (2015) propose nonparametric inverse optimization motivated by kernel methods in nonparametric regression. Chan et al. (2019) study the similarity between inverse LP and linear regression and propose a goodness-of-fit measure for inverse LP, analogous to the R-squared measure in regression.

Inverse LP has received particular attention among different types of inverse problems because of its potential to find an LP formulation—whether the true decision-generating system is linear or not—which is easy to solve while closely replicating the system. Although inverse optimization for more complex (e.g., nonconvex) forward problems might lead to a better fit, the resulting forward problems can be computationally

burdensome. Inverse LP has also been used for inferring LP formulations customized to different DMs or problem instances, which can replace a complex formulation in a personalized or distributed manner, for example, different LP cancer therapy planning formulations inferred for different types of patients, thus enabling personalized treatment modeling (Boutilier et al. 2015).

However, such potential of inverse LP often does not translate well into situations in which the observed decisions are subject to noise, measurement errors, and uncertainty. In particular, objective functions (or cost vectors) obtained by the existing inverse LP methods can be very sensitive to small changes in the data. For example, suppose that the DM's true decisions turn out slightly different from the observed ones because of measurement errors, or some of them are simply outliers because of the DM's inconsistent behavior. The cost vector inferred by the existing methods often changes substantially in response to such errors and outliers, leading to substantially different forward solutions; also, the forward solutions are often far removed from the observed decisions, making it hard to predict the performance of the inferred cost vector (see Section 2 for a detailed illustration). Such instability can limit the applicability of inverse LP in data-driven settings.

In this study, we propose a novel inverse LP framework that addresses the instability issues caused by noise, errors, and uncertainty in data. We formally define stability measures in inverse optimization and propose a new inverse model that improves on the previous methods. Inspired by least quantile linear regression, our model infers a cost vector based on a quantile statistic of optimality errors associated with the observed decisions. Furthermore, we aim to find a set of cost vectors that are guaranteed to generate solutions within a specified distance from the relevant subset of the data. The presentation of our method in this paper focuses on the setting in which the constraint matrix of the forward problem remains the same although observed decisions may vary. This setting can find various real-world applications, for example, diet problems in which nutritional factors for each food do not change and network optimization problems in which the geographical configuration (i.e., nodes and arcs) remains the same. We also discuss how this method can be extended when such assumptions do not hold.

1.1. Relevant Literature

Recent studies in inverse optimization focus on data-driven settings in which a large (potentially noisy) data set collected over a period of time or from many DMs is used for inferring the objective functions. Keshavarz et al. (2011) formulate an inverse model based on relaxed Karush–Kuhn–Tucker (KKT) conditions and impute a convex objective function that minimizes the KKT residuals with respect to the input solutions. Bertsimas et al. (2015) consider inverse variational inequality with noisy data and find model parameters that minimize the optimality gap associated with the data. Similarly, Esfahani et al. (2018) develop a distributionally robust inverse optimization model to infer an objective function from noisy data. Aswani et al. (2018) introduce the notion of risk consistency in inverse optimization and propose a model that finds an objective function that replicates the data in a statistically consistent manner. Inverse optimization has also been used for online learning with which the inferred objective function is updated adaptively as new data are available over time (Bärmann et al. 2017, Dong et al. 2018). Although these inverse convex programming frameworks can be specialized to inverse LPs, they are rather focused on generic convex programs with assumptions that preclude the aforementioned instability issues (e.g., strictly convex feasible regions) and, thus, are not designed to address these particular issues in inverse LP.

Although several recent inverse LP methods accommodate data that may not be optimal, these works focus on the development of closed-form solutions or efficient algorithms under often limiting assumptions (Chan et al. 2014, Ghobadi et al. 2018, Babier et al. 2021). As a result, how the obtained cost vectors actually work in the presence of data imperfection has received little attention. In particular, the previous studies exploit the polyhedral nature of the underlying forward LP and develop algorithms that find a cost vector orthogonal to one of the constraint vectors. Although such algorithms are efficient, the resulting cost vector is highly sensitive to outliers or small data shifts.

Our modeling approach to address the instability issues follows in spirit the line of work in the robust linear regression literature. Robust regression aims to infer a model that is stable against outliers or data shifts by increasing the stability measure of the fitted model, known as the breakdown point, which is defined as the fraction of the data that can be altered arbitrarily without making the model arbitrarily bad. Various robust regression techniques have been proposed, including least median of squares (Rousseeuw 1984), least trimmed squares (Rousseeuw and Van Driessen 2006, Hubert et al. 2008), and least quantile of squares methods (Bertsimas and Mazumder 2014). In these techniques, a parameter indicating the fraction of the data to be "trimmed" is prespecified based on the user's application-specific knowledge and preferences, and model-fitting is done by minimizing a certain quantile error statistic, excluding the residuals associated with data points that are deemed to be outliers.

1.2. Contributions and Organization of the Paper

Our contributions are as follows.

1. We introduce the notion of forward and inverse stability in inverse optimization and show that the previous inverse LP methods are often unstable in the face of data noise, uncertainties, and outliers. We then propose a new inverse LP method that improves on the previous methods in terms of inverse and forward stability measures and provides bounds for both measures.

2. We formulate the new inverse LP model as a large-scale MIP and formally characterize the set of all feasible cost vectors. By establishing a new connection between the inverse model and a class of biclique problems, we develop efficient algorithms that solve many smaller MIPs (often in parallel) instead of directly solving the large MIP. We show that one of the algorithms is exact with a condition that is easy to check, and other heuristics are useful for very large instances.

3. We demonstrate two settings in which the proposed inverse model is relevant: (i) the user observes decisions from an unknown system and aims to infer an LP formulation that can make new decisions similar to the observations; (ii) the user provides an LP formulation to the DMs as a signal and collects their decisions as a response, and these are then used for inferring their objective functions for the LP. For the former, we demonstrate our method in the diet recommendation problem to quantify an individual's diet preferences from noisy and inconsistent data. For the latter, we use the transshipment problem to infer costs perceived by the DMs from a sequence of noisy data sets.

The rest of the paper is organized as follows. In Section 2, we illustrate how the previous inverse LP can be sensitive to data noise and outliers and formalize the notion of stability in inverse optimization. In Section 3, we present the new inverse LP model that addresses the instability issues as well as its MIP reformulation. In Section 4, we discuss the connection between the proposed model and biclique problems and propose efficient solution approaches. In Section 5, we examine the performance of our model in terms of stability using various LP instances and demonstrate its use in the diet recommendation and transshipment problems. We conclude in Section 6. Unless otherwise stated, proofs are in the appendix (Appendix C). Throughout this paper, vectors are column vectors and $\mathbf{a}'$ denotes the transpose of vector $\mathbf{a}$, $\mathbf{e}$ denotes the vector of ones, $\mathrm{cone}(\cdot)$ denotes the set of conic combinations of given vectors, and $\mathrm{cone}_+(\cdot)$ denotes the set of strict conic combinations of given vectors.

2. Preliminaries

In this section, we first present the forward LP problem we consider and the previous inverse LP model. We then provide illustrative examples for the instability issues of the previous model and formally define a notion of stability in inverse LP.

2.1. Forward Linear Program

We consider the following forward optimization (FO) problem:

$$\mathbf{FO(c)}: \quad \underset{\mathbf{x}}{\text{minimize}} \, \{\mathbf{c}'\mathbf{x} \mid \mathbf{A}\mathbf{x} \geq \mathbf{b}\}, \tag{1}$$

where $\mathbf{c} \in \mathbb{R}^n, \mathbf{x} \in \mathbb{R}^n, \mathbf{A} \in \mathbb{R}^{m \times n}$, and $\mathbf{b} \in \mathbb{R}^m$. Let $\mathcal{I} = \{1, \ldots, m\}$ index the constraints, $\mathcal{J} = \{1, \ldots, n\}$ index the variables, and $\mathbf{a}^i \in \mathbb{R}^n$ be a (column) vector corresponding to the ith row of $\mathbf{A}$. Let $\mathcal{X}$ be the set of feasible solutions for the FO problem, assumed bounded, full-dimensional, and free of redundant constraints, and $\mathcal{X}_i = \{\mathbf{x} \in \mathcal{X} \mid \mathbf{a}^{i'}\mathbf{x} = b_i\}, i \in \mathcal{I}$. Without loss of generality, we assume that $\mathbf{a}^i$ for each $i \in \mathcal{I}$ is normalized a priori such that $\|\mathbf{a}^i\|_p = 1$ for some $p \geq 1$.

2.2. Previous Inverse Linear Programming Method

Consider a set of K data points (or observations) $\hat{\mathcal{X}} = \{\hat{\mathbf{x}}^1, \ldots, \hat{\mathbf{x}}^K\}$ with the index set $\mathcal{K} = \{1, \ldots, K\}$. We make no assumption on the feasibility or optimality of the observations for the forward problem (1). Previous inverse LP methods with suboptimal or infeasible observations aim to find a $\mathbf{c}$ vector that can generate a forward optimal solution that is closest to the observations $\{\hat{\mathbf{x}}^k\}_{k \in \mathcal{K}}$ under some distance metric, thus closely "reproducing" the data (Bertsimas et al. 2015, Aswani et al. 2018, Chan et al. 2019). In LP, using the general ℓ-norm as a distance metric, such an inverse model can be written as follows (Aswani et al. 2018, Babier et al. 2021):

$$\underset{\mathbf{c}, \{\boldsymbol{\epsilon}^k\}_{k \in \mathcal{K}}, \mathbf{y}}{\text{minimize}} \quad \sum_{k \in \mathcal{K}} \|\boldsymbol{\epsilon}^k\|_{\ell} \tag{2a}$$

$$\text{subject to} \quad \mathbf{A}'\mathbf{y} = \mathbf{c}, \tag{2b}$$

$$\mathbf{y} \geq \mathbf{0}, \tag{2c}$$

$$\mathbf{A}(\hat{\mathbf{x}}^k - \boldsymbol{\epsilon}^k) \geq \mathbf{b}, \quad \forall k \in \mathcal{K}, \tag{2d}$$

$$\mathbf{c}'(\hat{\mathbf{x}}^k - \boldsymbol{\epsilon}^k) = \mathbf{b}'\mathbf{y}, \quad \forall k \in \mathcal{K}, \tag{2e}$$

$$\|\mathbf{c}\|_p = 1, \tag{2f}$$

where $\ell \geq 1, p \geq 1$, and $\boldsymbol{\epsilon}^k$ denotes a perturbation vector for observation $\hat{\mathbf{x}}^k$. Given observations $\{\hat{\mathbf{x}}^k\}_{k \in \mathcal{K}}$, the problem finds a $\mathbf{c}$ vector that makes each perturbed solution $\hat{\mathbf{x}}^k - \boldsymbol{\epsilon}^k$ satisfy dual feasibility (2b) and (2c), primal feasibility (2d), and strong duality (2e), thus rendering it optimal, and the perturbations (i.e., "optimality errors" associated with the observations) are minimized in the ℓ-norm in the objective function (2a). The normalization constraint (2f) prevents a trivial, all-zero cost vector from being feasible.

Note that existing inverse convex programming models (e.g., Aswani et al. 2018) can be written equivalently as (2) when the underlying forward problem is

exactly (1). Although the preceding problem is nonconvex, recent works propose an efficient, exact algorithm for the problem by exploiting the solution structure (Chan et al. 2019, Babier et al. 2021), namely, that an optimal cost vector $\mathbf{c}^*$ is orthogonal to one of the hyperplanes defining polyhedron $\mathcal{X}$ (i.e., $\mathbf{c}^* = \mathbf{a}^i$ for some $i \in \mathcal{I}$).

2.3. Instability Issues in Inverse LP

When the data set $\hat{\mathcal{X}}$ contains noise or measurement errors, the previous inverse model (2) can be unstable in terms of both the cost vector $\mathbf{c}^*$ it produces as well as the forward solution that $\mathbf{c}^*$ generates (i.e., $\mathbf{x}^* \in \arg\min \mathbf{FO}(\mathbf{c}^*)$). The following examples illustrate such instability issues.

Example 1. Consider the forward LP $\text{minimize}_x \{\mathbf{c}'\mathbf{x} \mid \mathbf{Ax} \geq \mathbf{b}\}$, where $\mathbf{a}^1 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$, $\mathbf{a}^2 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$, $\mathbf{a}^3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, and $\mathbf{a}^4 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ are the rows of $\mathbf{A}$ (written as column vectors), and $\mathbf{b} = [-2.5, -2.5, 0, 0]'$ with four data points $\hat{\mathbf{x}}^1 = \begin{bmatrix} 2 \\ 2.3 \end{bmatrix}$, $\hat{\mathbf{x}}^2 = \begin{bmatrix} 2.2 \\ 2.3 \end{bmatrix}$, $\hat{\mathbf{x}}^3 = \begin{bmatrix} 2.2 \\ 2 \end{bmatrix}$, and $\hat{\mathbf{x}}^4 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ (see Figure 1(a)). The previous inverse model (2) with $\|\cdot\|_\ell = \|\cdot\|_\infty$ finds $\mathbf{c}^* = \mathbf{a}^1 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$ as the unique optimal cost vector with the objective value of 1.4 $\left(\epsilon^1 = \begin{bmatrix} 0 \\ -0.2 \end{bmatrix}, \epsilon^2 = \begin{bmatrix} 0 \\ -0.2 \end{bmatrix}, \epsilon^3 = \begin{bmatrix} 0 \\ -0.5 \end{bmatrix}, \text{ and } \epsilon^4 = \begin{bmatrix} 0 \\ -0.5 \end{bmatrix} \right)$. Suppose that $\hat{\mathbf{x}}^2$ and $\hat{\mathbf{x}}^3$ are shifted to $\begin{bmatrix} 2.3 \\ 2.2 \end{bmatrix}$ and $\begin{bmatrix} 2.3 \\ 1.9 \end{bmatrix}$, respectively (shown in gray). Even with these small shifts, the optimal cost vector changes from $\mathbf{c}^* = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$ (which now leads to the objective value of 1.6) to a substantially different vector $\mathbf{c}^* = \mathbf{a}^2 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$ with the objective value of 1.4 $\left(\epsilon^1 = \begin{bmatrix} -0.5 \\ 0 \end{bmatrix}, \epsilon^2 = \begin{bmatrix} -0.2 \\ 0 \end{bmatrix}, \epsilon^3 = \begin{bmatrix} -0.2 \\ 0 \end{bmatrix}, \text{ and } \epsilon^4 = \begin{bmatrix} -0.5 \\ 0 \end{bmatrix} \right)$.

Example 2. Consider the same initial data points from Example 1. Suppose that an outlier $\hat{\mathbf{x}}^5 = \begin{bmatrix} 2.2 \\ 0.3 \end{bmatrix}$ is introduced (see Figure 1(b)). The optimal cost vector then changes from $\mathbf{c}^* = \mathbf{a}^1 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$ to a substantially different vector $\mathbf{c}^* = \mathbf{a}^2 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$ with the objective value of 1.9 $\left(\epsilon^1 = \begin{bmatrix} -0.5 \\ 0 \end{bmatrix}, \epsilon^2 = \begin{bmatrix} -0.3 \\ 0 \end{bmatrix}, \epsilon^3 = \begin{bmatrix} -0.3 \\ 0 \end{bmatrix}, \epsilon^4 = \begin{bmatrix} -0.5 \\ 0 \end{bmatrix}, \text{ and } \epsilon^5 = \begin{bmatrix} -0.3 \\ 0 \end{bmatrix} \right)$. The previous solution now has the objective value of 1.4 + $\left\| \begin{bmatrix} 0 \\ -2.2 \end{bmatrix} \right\|_\infty = 3.6$ and, hence, is no longer optimal.

Example 3. Consider the five data points from Example 2 with which the optimal cost vector is $\mathbf{c}^* = \mathbf{a}^2 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$. Although a desirable forward solution is supposed to be close to the relevant subset of the data without the outlier, Figure 1(c) shows that using this cost vector for the forward LP can lead to a solution $\mathbf{x}^*$ that is far from the majority of the data set.

Examples 1 and 2 show that the previous inverse model is sensitive to outliers or small data shifts, which we refer to as being *inverse-unstable*. On the other hand, Example 3 shows that solving the forward problem with a cost vector from the previous model can lead to a solution that is (unexpectedly) far from the relevant subset of the data, which we refer to as being *forward-unstable*.

We note that these instability issues are common in general inverse LP settings. For example, if a data-generating system produces errors 10% of the time, the inferred cost vector can still be heavily dragged toward these errors spread far from the majority of the data regardless of the size of the data set, leading to a cost vector that would have not been chosen if only the pristine data (the remaining 90%) was used. Also, when small shifts occur in multiple data points according to the same distribution (e.g., the DM's behavior changes gradually over time), the cost vector can also suddenly change to a substantially different vector, leading to a substantially different forward solution. In general, these instability issues depend heavily on the spatial distribution of the data in relation to polyhedron $\mathcal{X}$, for example, the distance between each data point and each facet of $\mathcal{X}$.

Figure 1. (Color online) Instability Issues of the Previous Inverse LP Method
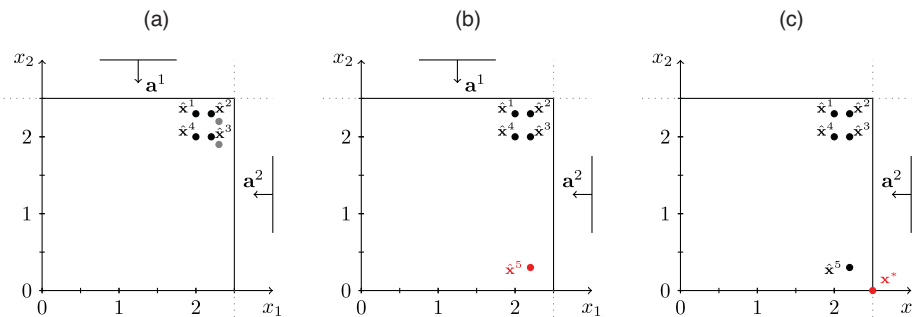
Notes. (a) Inverse instability resulting from data shift. (b) Inverse instability resulting from outliers. (c) Forward instability.

2.4. Stability Measures for Inverse LP

In this section, we formally define a notion of stability in inverse LP and propose measures that we use to assess the stability of an inverse LP model. Given a data set $\hat{\mathcal{X}}$ and a certain inverse LP model, let $\mathcal{C}(\hat{\mathcal{X}})$ be the set of cost vectors obtained by the model with $\hat{\mathcal{X}}$ as input.

First, we propose to measure inverse stability of an inverse LP model (or solutions to the model) by the minimum tolerable data shift in $\hat{\mathcal{X}}$ until the model loses all of its initial solutions in $\mathcal{C}(\hat{\mathcal{X}})$. We measure such tolerable shift via some distance function $d(\hat{\mathcal{X}}, \tilde{\mathcal{X}})$, where $\tilde{\mathcal{X}}$ denotes data shifted from $\hat{\mathcal{X}}$; for example, $d(\hat{\mathcal{X}}, \tilde{\mathcal{X}}) = \max\{\|\hat{\mathbf{x}} - \tilde{\mathbf{x}}\|_\ell \mid \hat{\mathbf{x}} \in \hat{\mathcal{X}}, \tilde{\mathbf{x}} \in \tilde{\mathcal{X}}\}$ and $d(\hat{\mathcal{X}}, \tilde{\mathcal{X}}) = \sum_{k \in \mathcal{K}} \|\hat{\mathbf{x}}^k - \tilde{\mathbf{x}}^k\|_\ell, \ell \geq 1$, if $\hat{\mathcal{X}}$ and $\tilde{\mathcal{X}}$ are in one-to-one correspondence.

Definition 1 (Inverse Stability). Given some distance function d and data set $\hat{\mathcal{X}}$, *inverse stability* of an inverse LP model (or cost vectors obtained by the model) is measured by

$$\min_{\tilde{\mathcal{X}}}\{d(\hat{\mathcal{X}}, \tilde{\mathcal{X}}) \mid \mathcal{C}(\hat{\mathcal{X}}) \cap \mathcal{C}(\tilde{\mathcal{X}}) = \emptyset\}. \quad (3)$$

That is, given two distinct sets of cost vectors $\mathcal{C}_1(\hat{\mathcal{X}})$ and $\mathcal{C}_2(\hat{\mathcal{X}})$ obtained by two inverse models (models 1 and 2, respectively) with the same input $\hat{\mathcal{X}}$, model 1 $(\mathcal{C}_1(\hat{\mathcal{X}}))$ is said to be more inverse-stable than model 2 $(\mathcal{C}_2(\hat{\mathcal{X}}))$ if $\min_{\tilde{\mathcal{X}}}\{d(\hat{\mathcal{X}}, \tilde{\mathcal{X}}) \mid \mathcal{C}_1(\hat{\mathcal{X}}) \cap \mathcal{C}_1(\tilde{\mathcal{X}}) = \emptyset\} > \min_{\tilde{\mathcal{X}}}\{d(\hat{\mathcal{X}}, \tilde{\mathcal{X}}) \mid \mathcal{C}_2(\hat{\mathcal{X}}) \cap \mathcal{C}_2(\tilde{\mathcal{X}}) = \emptyset\}$.

This definition of inverse stability suggests that, ideally, a stable inverse model should maintain some cost vectors when reasonably small changes occur in the data as quantified by the preceding measure. Note that this measure is analogous to the stability measure in regression, that is, the finite-sample breakdown point, defined as the fraction of the data that can be altered without spoiling the inferred model completely (Yohai 1987).

Next, we define forward stability in inverse LP. Suppose we select $\hat{\mathbf{c}} \in \mathcal{C}(\hat{\mathcal{X}})$ and find a set of forward optimal solutions $\mathcal{X}^*(\hat{\mathbf{c}}) = \text{argmin } \mathbf{FO}(\hat{\mathbf{c}})$. How *unstable* this cost vector $\hat{\mathbf{c}}$ can be is assessed by how far a forward solution $\mathbf{x} \in \mathcal{X}^*(\hat{\mathbf{c}})$ can be from the given observations $\hat{\mathcal{X}}$, that is, the worst-case distance between $\mathcal{X}^*(\hat{\mathbf{c}})$ and $\hat{\mathcal{X}}$ via a distance function defined similarly as before.

Definition 2 (Forward Stability). Given some distance function d, data set $\hat{\mathcal{X}}$ and cost vector $\hat{\mathbf{c}}$, we measure how forward-unstable the cost vector $\hat{\mathbf{c}}$ is by

$$\max_{\mathbf{x} \in \mathcal{X}^*(\hat{\mathbf{c}})}\{d(\hat{\mathcal{X}}, \mathbf{x})\}. \quad (4)$$

That is, given two distinct cost vectors $\hat{\mathbf{c}}^1$ and $\hat{\mathbf{c}}^2$, $\hat{\mathbf{c}}^1$ is said to be more *forward-stable* than $\hat{\mathbf{c}}^2$ if $\max_{\mathbf{x} \in \mathcal{X}^*(\hat{\mathbf{c}}^1)}\{d(\hat{\mathcal{X}}, \mathbf{x})\} < \max_{\mathbf{x} \in \mathcal{X}^*(\hat{\mathbf{c}}^2)}\{d(\hat{\mathcal{X}}, \mathbf{x})\}$.

Definition 2 suggests that, to guarantee a reasonable level of forward stability, an inverse model should identify a cost vector that is guaranteed to produce an optimal solution that is reasonably close to the observations.

The inverse and forward stability measures often conflict with each other. For example, a cost vector whose components are all zeros, if feasible for a certain inverse model (e.g., when the normalization constraint is not used; Dong et al. 2018, Naghavi et al. 2019), can be highly inverse-stable as some arbitrary data set $\tilde{\mathcal{X}} \subset \mathcal{X}$ can still be optimal for the forward problem with respect to such an all-zero cost vector (e.g., any solutions on the boundary of $\mathcal{X}$), yet using this vector for the forward problem likely leads to a solution far from the given data, that is, highly forward-unstable. Also, one might attempt to include as many cost vectors as possible in $\mathcal{C}(\hat{\mathcal{X}})$ just to make it less vulnerable to data shifts (i.e., more inverse-stable), yet which of these vectors is truly forward-stable then becomes less clear. Another challenge is that neither of these measures lends itself to a tractable inverse optimization problem. In the next section, we provide a tractable inverse LP framework that improves both inverse and forward stability as quantified by these measures.

3. Models

In this section, we first propose a new inverse LP model that improves the inverse stability of the previous model (2). We then analyze the solution structure of the model, which we exploit to reformulate the model as an MIP and characterize the set of all feasible cost vectors. Finally, we introduce a specific objective function for the MIP that can serve as a proxy for the forward stability measure and leads to a cost vector with improved forward stability.

3.1. Inverse LP for Quantile Statistics: Improving Inverse Stability

As discussed, inverse instability is caused by the vulnerability of inferred cost vectors to data shifts or outliers. Our modeling strategy to address this is inspired by the close relationship between the least squares method in regression and inverse optimization: the previous inverse model (2) is similar to the least squares method in that it fits the LP model by minimizing the sum of optimality errors (Chan et al. 2019). In regression, one way to address data imperfection is to minimize the θ quantile statistic where $\theta \in [0, 1]$ (i.e., $(\theta \times 100)$-th percentile) instead of the sum of the squared errors, which is known as the least quantile of squares method (Koenker and Hallock 2001, Bertsimas and Mazumder 2014) or least trimmed quantile regression (Rousseeuw 1984, Rousseeuw and Van Driessen 2006).

We adopt this idea and propose a generalized inverse framework in which a cost vector is inferred based on the θ quantile optimality error, that is, associated with a data point that induces the $\lceil\theta K\rceil$th smallest optimality error with respect to the cost vector. Instead of minimizing this θ quantile error as was done in Bertsimas and Mazumder (2014), we find a set of cost vectors such that the $\lceil\theta K\rceil$th smallest optimality error is no greater than a certain threshold τ. That is, such cost vectors render a relevant subset of the data (of cardinality $\lceil\theta K\rceil$) within the threshold optimality error. This problem, which we call the quantile inverse optimization (QIO) problem, can be written as follows:

$$\mathbf{QIO}(\mathcal{K},\tau,\theta): \quad \underset{\mathbf{c},\{\epsilon^k\}_{k\in\mathcal{K}},\mathbf{y},\mathcal{S}}{\text{minimize}} \quad 0 \tag{5a}$$

$$\text{subject to} \quad \mathbf{A'y}=\mathbf{c}, \tag{5b}$$

$$\mathbf{y}\geq\mathbf{0}, \tag{5c}$$

$$\mathbf{A}(\hat{\mathbf{x}}^k-\epsilon^k)\geq\mathbf{b}, \quad \forall k\in\mathcal{S}, \tag{5d}$$

$$\mathbf{c'}(\hat{\mathbf{x}}^k-\epsilon^k)=\mathbf{b'y}, \quad \forall k\in\mathcal{S}, \tag{5e}$$

$$\|\epsilon^k\|_\ell\leq\tau, \quad \forall k\in\mathcal{S}, \tag{5f}$$

$$|\mathcal{S}|\geq\theta K, \tag{5g}$$

$$\mathcal{S}\subseteq\mathcal{K}, \tag{5h}$$

$$\|\mathbf{c}\|_p=1. \tag{5i}$$

Constraints (5b) and (5c) represent dual feasibility, and (5d) and (5e) enforce primal feasibility and strong duality associated with the perturbed solutions $\hat{\mathbf{x}}^k-\epsilon^k$ for a subset of the data $\mathcal{S}\subseteq\mathcal{K}$ whose cardinality is enforced to be no less than θK by (5g). Constraint (5f) then ensures that optimality error for each chosen observation is within τ. Clearly, these constraints ensure that the $\lceil\theta K\rceil$th smallest optimality error is no greater than τ. Note that this problem is written as a feasibility problem; we later introduce an objective function that leads to a specific subset of the feasible cost vectors (see Section 3.2). We call cost vectors that are feasible for (5b)–(5i) "inverse-feasible" cost vectors. Similar to quantile-based regression, the choice of θ depends on the application context or user preference; for example, θ can be chosen based on the user's belief about the fraction of outliers and can also be adjusted post hoc depending on the result from the model. Or it can be set to 50% if one is interested in the median error-based inverse problem. This problem is referred to as $\mathbf{QIO}(\mathcal{K},\tau,\theta)$ or $\mathbf{QIO}(\hat{\mathcal{X}},\tau,\theta)$ interchangeably, depending on which is more convenient for the context.

A strong advantage of this modeling framework is that it generalizes the previous model (2); given the same data set $\hat{\mathcal{X}}$, there exists τ, say $\bar{\tau}$, such that $\mathbf{QIO}(\hat{\mathcal{X}},\bar{\tau},\theta)$ with $\theta=100\%$ produces the same set of feasible cost vectors as (2). Furthermore, given such $\bar{\tau}$, as θ decreases, the set of feasible cost vectors from $\mathbf{QIO}(\hat{\mathcal{X}},\bar{\tau},\theta)$ contains the cost vectors from the previous model because the strong duality constraint (5e) is allowed to hold with fewer data points. That is, in the presence of data shift or additional outliers, $\mathbf{QIO}(\hat{\mathcal{X}},\bar{\tau},\theta)$ does not lose all of its initial inverse-feasible solutions until the previous model does; that is, it is at least as stable as the previous model as quantified by inverse stability measure (3). Note that it is straightforward to find such $\bar{\tau}$: we can either set it to a sufficiently large number (e.g., based on Appendix A.1) or solve the previous model (2) via an efficient algorithm in the literature and set $\bar{\tau}$ to be no less than the maximum optimality error across all data points obtained by Model (2). The following result further generalizes this observation and shows that, for any given $\hat{\mathcal{X}}$ and τ, the inverse stability measure (3) is nondecreasing as θ decreases.

Proposition 1. *Let $\hat{\mathcal{C}}(\theta)$ be the set of inverse-feasible cost vectors for $\mathbf{QIO}(\hat{\mathcal{X}},\tau,\theta)$. Given $\hat{\mathcal{X}}$ and τ, the inverse stability measure (3) for $\hat{\mathcal{C}}(\theta)$ is nondecreasing as θ decreases.*

We later further show that a solution to $\mathbf{QIO}(\hat{\mathcal{X}},\tau,\theta)$ provides a lower bound on the inverse stability measure. In the following, we first analyze the solution structure of $\mathbf{QIO}(\hat{\mathcal{X}},\tau,\theta)$.

3.1.1. Solution Structure and MIP Reformulation.
The QIO problem is nonconvex because of Constraint (5e) and, thus, is hard to solve. We analyze the solution structure of this problem, which leads to an MIP reformulation and a lower bound on the inverse stability measure.

Proposition 2. *If $\mathbf{QIO}(\mathcal{K},\tau,\theta)$ is feasible, then there exists a feasible $\mathbf{c}$ for $\mathbf{QIO}(\mathcal{K},\tau,\theta)$ such that $\mathbf{c}=\mathbf{a}^i$ for some $i\in\mathcal{I}$.*

Proposition 2 implies that the feasibility of the QIO model can be checked by evaluating at most m constraint vectors, that is, $\mathbf{a}^i, i=1,\dots,m$. For example, revisiting Example 1 (Figure 1(a)), for the QIO problem with $\|\cdot\|_\ell=\|\cdot\|_\infty$, $\theta=0.5$, and $\tau=0.3$, both $\mathbf{c}=\mathbf{a}^1$ and $\mathbf{c}=\mathbf{a}^2$ are feasible: $\mathbf{c}=\mathbf{a}^1$ leads to $\|\epsilon^1\|_\infty=\|\epsilon^2\|_\infty=0.2$ and $\mathbf{c}=\mathbf{a}^2$ leads to $\|\epsilon^2\|_\infty=\|\epsilon^3\|_\infty=0.3$, both satisfying the threshold $\tau=0.3$. Proposition 2 also suggests that, if there is no $\mathbf{c}=\mathbf{a}^i$ feasible for (5) for any $i\in\mathcal{I}$, then there exists no cost vector that can make at least $\lceil\theta K\rceil$ observations within τ-optimality. In this case, users could decrease θ or increase τ to make the model feasible. Or τ can be set by adding a reasonable margin to the minimum possible value of τ that keeps the model feasible (see Appendix A.1 for more details). With an appropriate threshold τ, the set of inverse-feasible cost vectors that are orthogonal to some hyperplanes

defining $\mathcal{X}$ can be found efficiently by evaluating each hyperplane i (see Algorithm 1).

Algorithm 1 (Finding Inverse-Feasible Solutions of the Form $\mathbf{c} = \mathbf{a}^i$, $i \in \mathcal{I}$)

  Result: $\mathcal{C} = \{\mathbf{a}^i\}_{i \in \mathcal{A}}$
  Input: $\mathbf{A}, \mathbf{b}, \hat{\mathcal{X}}, \tau, \theta$
  1. $\mathcal{S}_i \leftarrow \emptyset \ \forall i \in \mathcal{I}$; $\mathcal{A} \leftarrow \emptyset$
  2. **for** $i \in \mathcal{I}$ **do**
  3.   **for** $k \in \mathcal{K}$ **do**
  4.   **if** $\exists \mathbf{x} \in \mathcal{X}_i$ such that $\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell \leq \tau$ **then**
  5. $\mathcal{S}_i \leftarrow \mathcal{S}_i \cup \{k\}$;
  6.   **end**
  7.   **end**
  8.   **if** $|\mathcal{S}_i| \geq \theta K$ **then**
  9. $\mathcal{A} \leftarrow \mathcal{A} \cup \{i\}$
  10.   **end**
  11. **end**

Note that Algorithm 1 returns the set of $\mathbf{a}^i$'s that are inverse-feasible, that is, $\mathcal{A}$, which is only a subset of the set of all inverse-feasible solutions. We extend the results in Proposition 2 and Algorithm 1 and derive the following result, which further identifies other inverse-feasible cost vectors.

Lemma 1. *If there exists $\bar{\mathbf{x}}^k \in \cap_{i \in \bar{\mathcal{A}}} \mathcal{X}_i$ for some set $\bar{\mathcal{A}} \subseteq \mathcal{A}$ such that $\|\bar{\mathbf{x}}^k - \hat{\mathbf{x}}^k\|_\ell \leq \tau$ for all $k \in \bar{\mathcal{S}} \subseteq \mathcal{K}$, where $|\bar{\mathcal{S}}| \geq \theta K$, then $\mathbf{c} \in \mathrm{cone}(\{\mathbf{a}^i\}_{i \in \bar{\mathcal{A}}})$ such that $\|\mathbf{c}\|_p = 1$ is inverse-feasible.*

Lemma 1 suggests that we can check each subset $\bar{\mathcal{A}}$ of $\mathcal{A}$ to see if a conic combination of $\mathbf{a}^i$'s for $i \in \bar{\mathcal{A}}$ is also inverse-feasible. Using this observation, we propose the following MIP:

$$\underset{\mathbf{v}, \mathbf{u}, \{\boldsymbol{\epsilon}^k\}_{k \in \mathcal{K}}}{\mathrm{minimize}} \quad 0 \tag{6a}$$

$$\text{subject to} \quad b_i \leq \mathbf{a}^{i\prime}(\hat{\mathbf{x}}^k - \boldsymbol{\epsilon}^k) \leq b_i + M_1(1 - v_i),$$
$$\forall i \in \mathcal{I}, \ \forall k \in \mathcal{K}, \tag{6b}$$

$$\|\boldsymbol{\epsilon}^k\|_\ell \leq \tau + M_2(1 - u_k), \quad \forall k \in \mathcal{K}, \tag{6c}$$

$$\sum_{k=1}^{K} u_k \geq \theta K, \tag{6d}$$

$$v_i, u_k \in \{0, 1\}, \quad \forall i \in \mathcal{I}, \ \forall k \in \mathcal{K}, \tag{6e}$$

where $v_i = 1$ if $\mathbf{a}^{i\prime}(\hat{\mathbf{x}}^k - \boldsymbol{\epsilon}^k) \geq b_i$ holds with equality (i.e., $\hat{\mathbf{x}}^k - \boldsymbol{\epsilon}^k \in \mathcal{X}_i$) for all $k \in \mathcal{K}$ and $v_i = 0$ otherwise, and $u_k = 1$ if observation $\hat{\mathbf{x}}^k$ is "chosen" and $u_k = 0$ otherwise. Parameters M_1 and M_2 denote sufficiently large constants; Appendix A.2 shows how to find appropriate values for them. In Formulation (6), inverse-feasible $\mathbf{a}^i$'s are identified by letting $v_i = 1$; the hypothesis of Lemma 1 is then explicitly written as constraints to further identify a conic combination of the chosen $\mathbf{a}^i$'s that is also inverse-feasible (the set $\mathcal{I}$ in (6b) and (6e) can be replaced by $\mathcal{A}$ without losing any feasible solution because v_i can be one only for $i \in \mathcal{A} \subseteq \mathcal{I}$ anyway; we use $\mathcal{I}$ in (6) and subsequent formulations for

notational clarity). The following result establishes the equivalence between Problem (6) and $\mathbf{QIO}(\mathcal{K}, \tau, \theta)$ in terms of the set of achievable inverse-feasible cost vectors given the same data set $\hat{\mathcal{X}}$.

Theorem 1. *A cost vector $\mathbf{c}$ is feasible for $\mathbf{QIO}(\mathcal{K}, \tau, \theta)$ if and only if there exists a feasible solution $(\bar{\mathbf{v}} \neq \mathbf{0}, \bar{\mathbf{u}}, \{\bar{\boldsymbol{\epsilon}}^k\}_{k \in \mathcal{K}})$ for Model (6) such that $\mathbf{c} \in \mathrm{cone}(\{\mathbf{a}^i\}_{i:\bar{v}_i=1})$ and $\|\mathbf{c}\|_p = 1$.*

Thus, an inverse-feasible cost vector for $\mathbf{QIO}(\mathcal{K}, \tau, \theta)$ can be obtained by finding a feasible solution $(\bar{\mathbf{v}}, \bar{\mathbf{u}}, \{\bar{\boldsymbol{\epsilon}}^k\}_{k \in \mathcal{K}})$ for Model (6) and creating a conic combination of $\mathbf{a}^i$'s for i such that $\bar{v}_i = 1$. In fact, without having to normalize $\mathbf{c}$ post hoc, for any conic combination of such $\mathbf{a}^i$'s, there is an inverse-feasible cost vector for $\mathbf{QIO}(\mathcal{K}, \tau, \theta)$ that generates the same forward optimal solutions. That is, if we let $\mathcal{C}_{QIO}$ be the set of all inverse-feasible $\mathbf{c}$ vectors for $\mathbf{QIO}(\mathcal{K}, \tau, \theta)$ and $\mathcal{C}_{MIP} = \bigcup_{\mathbf{v} \in \hat{\mathcal{V}}} \mathrm{cone}(\{\mathbf{a}^i\}_{i:v_i=1})$, where $\hat{\mathcal{V}}$ is the set of feasible nonzero $\mathbf{v}$ vectors for Model (6), we have $\bigcup_{\mathbf{c} \in \mathcal{C}_{QIO}} \mathrm{argmin}\, \mathbf{FO}(\mathbf{c}) = \bigcup_{\mathbf{c} \in \mathcal{C}_{MIP}} \mathrm{argmin}\, \mathbf{FO}(\mathbf{c})$. The following result formally characterizes the set of all inverse-feasible solutions for $\mathbf{QIO}(\mathcal{K}, \tau, \theta)$ (proof is straightforward from the proof of Theorem 1 and, thus, is omitted).

Corollary 1. *Let $\Pi(\hat{\mathcal{X}})$ denote the feasible region of Model (6). Then, the set of all inverse-feasible cost vectors, $\hat{\mathcal{C}}$, can be characterized by $\hat{\mathcal{V}} = \mathrm{Proj}_\mathbf{v}(\Pi(\hat{\mathcal{X}})) \backslash \{\mathbf{0}\}$. That is, $\hat{\mathcal{C}} = \bigcup_{\mathbf{v} \in \hat{\mathcal{V}}} \mathrm{cone}(\{\mathbf{a}^i\}_{i:v_i=1})$.*

3.1.2. Lower Bound on the Inverse Stability Measure. Finally, we show that a solution to (6) provides a lower bound on the inverse stability measure (3) for $\mathbf{QIO}(\mathcal{K}, \tau, \theta)$. To do so, we make the following assumptions: (i) data noise in $\hat{\mathcal{X}}$ is in the form of point-wise shift, that is, a shifted data point is expressed as $\tilde{\mathbf{x}}^k = \hat{\mathbf{x}}^k + \boldsymbol{\delta}^k$ for each $k \in \mathcal{K}$; (ii) $d(\hat{\mathcal{X}}, \tilde{\mathcal{X}}) = \sum_{k \in \mathcal{K}} \|\hat{\mathbf{x}}^k - \tilde{\mathbf{x}}^k\|_\ell$ for some $\ell \geq 1$.

Proposition 3. *Let ξ^* be the inverse stability value for $\mathbf{QIO}(\mathcal{K}, \tau, \theta)$. Let $(\bar{\mathbf{v}} \neq 0, \bar{\mathbf{u}}, \{\bar{\boldsymbol{\epsilon}}^k\}_{k \in \mathcal{K}})$ be a feasible solution to (6), $\bar{\mathcal{I}} = \{i \in \mathcal{I} \mid \bar{v}_i = 1\}$, $d_{ik}^* = \min_{\mathbf{x} \in \mathcal{X}_i}\{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\}$, $i \in \bar{\mathcal{I}}, k \in \mathcal{K}$, and $d_{i[k]}^*$ denote the kth largest value of d_{ik}^*, $k \in \mathcal{K}$. Then, we have $\max_{i \in \bar{\mathcal{I}}}\{\sum_{k=1}^{\lfloor (1-\theta)K \rfloor + 1} \max(0, \tau - d_{i[k]}^*)\} \leq \xi^*$.*

Given $\mathcal{K}$ and τ, we note that the lower bound increases as θ decreases, which reinforces the role of θ in increasing inverse stability. Our numerical results show that the lower bound is nontrivial and close to the numerically estimated inverse stability measure.

Although Problem (6) leads to the set of cost vectors with improved inverse stability, which cost vector among them to use for the forward problem is an important

consideration to improve forward stability. In the next section, we introduce an objective function for Problem (6) to find a **c** vector with improved forward stability.

3.2. Finding the Maximal Dimension Inverse-Feasible Set: Improving Forward Stability

Because it is hard to model forward stability measure (4) as a tractable objective function, we propose a surrogate, tractable function that can be used as an objective function for the QIO model. Recall that the forward stability measure for a cost vector **c** represents the worst-case distance between $\mathcal{X}^*(\mathbf{c}) = \text{argmin}_\mathbf{x}$ **FO(c)** and the data. We compute forward stability only in terms of the data points that are actually used by the model and consider **c** to be forward-stable if the worst-case distance is no greater than the error threshold τ for the relevant subset of the data of cardinality $\lceil \theta K \rceil$ (i.e., excluding those deemed to be outliers). We first make the following observation.

Proposition 4. *If a feasible solution $(\bar{\mathbf{v}}, \bar{\mathbf{u}}, \{\bar{\epsilon}^k\}_{k \in \mathcal{K}})$ for Problem (6) satisfies $\sum_{i \in \mathcal{I}} \bar{v}_i = n$, then $\bar{\mathbf{c}} \in \text{cone}_+(\{\mathbf{a}^i\}_{i:\bar{v}_i=1})$ satisfies $\max_{\mathbf{x} \in \mathcal{X}^*(\bar{\mathbf{c}})} \{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\} \leq \tau, \forall k : \bar{u}_k = 1$.*

Proposition 4 implies that, if **c** is a strict conic combination of n $\mathbf{a}^i$'s chosen by (6) (i.e., such that $\bar{v}_i = 1$), then **FO(c)** has a unique optimal solution that is within τ-distance from all chosen data points (Mangasarian 1979, Tavaslioğlu et al. 2018), and thus, the forward stability measure is guaranteed to be no greater than the prespecified threshold. Furthermore, the following result shows that, given a set of data points chosen to be nonoutliers, as we assign more ones to the components of a solution **v** in Problem (6) (i.e., adding more basis vectors to the conic combination), the resulting **c** vector can improve forward stability.

Proposition 5. *Given $\hat{\mathcal{X}}$, let $(\mathbf{v}^1, \bar{\mathbf{u}})$ and $(\mathbf{v}^2, \bar{\mathbf{u}})$ be two distinct solutions feasible for Problem (6). Let $\mathcal{C}_1 = \text{cone}(\{\mathbf{a}^i\}_{i:v_i^1=1})$ and $\mathcal{C}_2 = \text{cone}(\{\mathbf{a}^i\}_{i:v_i^2=1})$. If $\mathbf{v}^1 \geq \mathbf{v}^2$, then for any $\mathbf{c}^2 \in \mathcal{C}_2$ there exists $\mathbf{c}^1 \in \mathcal{C}_1$ such that $\max_{\mathbf{x} \in \mathcal{X}^*(\mathbf{c}^1)} \{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\} \leq \max_{\mathbf{x} \in \mathcal{X}^*(\mathbf{c}^2)} \{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\}, \forall k : \bar{u}_k = 1$; that is, there exists $\mathbf{c}^1 \in \mathcal{C}_1$ that is at least as forward-stable as any $\mathbf{c}^2 \in \mathcal{C}_2$.*

We also show that adding more basis vectors to the conic combination also improves inverse stability.

Proposition 6. *Given $\hat{\mathcal{X}}$, let $\mathbf{v}^1$ and $\mathbf{v}^2$ be two distinct solutions feasible for Problem (6). Let $\mathcal{C}_1 = \text{cone}(\{\mathbf{a}^i\}_{i:v_i^1=1})$ and $\mathcal{C}_2 = \text{cone}(\{\mathbf{a}^i\}_{i:v_i^2=1})$. If $\mathbf{v}^1 \geq \mathbf{v}^2$, then $\min_{\tilde{\mathcal{X}}} \{d(\hat{\mathcal{X}}, \tilde{\mathcal{X}}) | \mathcal{C}_1 \cap \mathcal{C}(\tilde{\mathcal{X}}) = \emptyset\} \geq \min_{\tilde{\mathcal{X}}} \{d(\hat{\mathcal{X}}, \tilde{\mathcal{X}}) | \mathcal{C}_2 \cap \mathcal{C}(\tilde{\mathcal{X}}) = \emptyset\}$, where $\mathcal{C}(\tilde{\mathcal{X}})$ denotes the set of inverse-feasible cost vectors for $\tilde{\mathcal{X}}$; that is, $\mathcal{C}_1$ is at least as inverse-stable as $\mathcal{C}_2$.*

Propositions 4–6 imply that we can add the objective function of maximizing the number of nonzero

v_i's to Problem (6) to find which cost vector to eventually use for the forward problem to further improve forward stability without compromising inverse stability. This leads us to the following mixed-integer quantile inverse optimization (MQIO) model:

$$\mathbf{MQIO}(\mathcal{K}, \tau, \theta): \quad \underset{\mathbf{v}, \mathbf{u}, \{\epsilon^k\}_{k \in \mathcal{K}}}{\text{maximize}} \left\{ \sum_{i \in \mathcal{I}} v_i \,\middle|\, (6b)-(6e) \right\}. \quad (7)$$

Our numerical results show that, even when the optimal value of (7) is not n but close to n, the MQIO model is more forward-stable than the previous inverse model. Computing forward stability is inherently challenging as it involves finding the maximum distance between the data set and some face of polyhedron $\mathcal{X}$. An upper bound for the forward-stability measure may be computed if some geometric information about the polyhedron is given, for example, the minimum volume ellipsoid or other size measure for some faces of $\mathcal{X}$, yet obtaining such information itself is not easy in general. In Appendix A.4, we show that an optimal solution to the MQIO model leads to an upper bound on the forward stability measure if such information is available.

Appendix A.5 provides extensions to the MQIO model that can accommodate situations in which $(\mathbf{A}, \mathbf{b})$ varies over different data points, $(\mathbf{A}, \mathbf{b})$ is subject to uncertainty, or the cost vector is constrained. The MQIO problem is a large-scale MIP in general and, thus, is computationally challenging. In the next section, we provide efficient solution approaches by exploiting the problem structure.

4. Solution Approaches

In this section, we develop efficient solution approaches to the MQIO problem. In Section 4.1, we first show that the MQIO problem is NP-hard by establishing its equivalence to a biclique problem. We then provide a formal biclique representation of a feasible solution to the QIO (hence, MQIO) problem in Section 4.2, which indicates that algorithms designed for biclique problems can be adapted to solve the MQIO problem. Finally in Section 4.3, utilizing the biclique representation, we propose an exact algorithm and its heuristic variants for the MQIO problem in which K smaller MIPs (or relaxations of them) are solved to construct a bipartite graph that leads to a feasible solution to the MQIO problem.

4.1. Problem Complexity and Connection to Biclique Problems

We show that $\mathbf{MQIO}(\mathcal{K}, \tau, \theta)$ is NP-hard by establishing its connection to biclique problems, which we later use to develop alternative algorithms for the problem. To do so, we first show that there exists an optimal

solution to $\mathbf{MQIO}(\mathcal{K}, \tau, \theta)$ with exactly $\lceil \theta K \rceil$ observations selected.

Lemma 2. *If $\mathbf{MQIO}(\mathcal{K}, \tau, \theta)$ is feasible, then there exists an optimal solution $(\mathbf{v}^*, \mathbf{u}^*, \{\boldsymbol{\epsilon}^{k*}\}_{k\in\mathcal{K}})$ such that $\sum_{k=1}^{K} u_k^* = \lceil \theta K \rceil$.*

Proposition 7. *The problem $\mathbf{MQIO}(\mathcal{K}, \tau, \theta)$ is NP-hard.*

To prove Proposition 7, we first introduce the maximum κ-subset intersection (MSI) problem. Consider a ground set $\mathcal{G} = \{g_1, \ldots, g_N\}$ and a set of its subsets $\mathscr{R} = \{\mathcal{R}_1, \ldots, \mathcal{R}_Q\}$, that is, $\mathcal{R}_{\mathbf{q}} \subseteq \mathcal{G}$, $\mathbf{q} = 1, \ldots, Q$. Given a positive integer κ, MSI finds κ subsets in $\mathscr{R}$ whose intersection has maximum cardinality. MSI problems are known to be NP-hard (Dawande et al. 2001, Xavier 2012).

Proof of Proposition 7. Consider the following feasibility problem for each $k \in \mathcal{K}$.

$$\underset{\mathbf{v}, \boldsymbol{\epsilon}}{\text{maximize}} \quad 0 \tag{8a}$$

$$\text{subject to} \quad b_i \leq \mathbf{a}^{i'}(\hat{\mathbf{x}}^k - \boldsymbol{\epsilon}) \leq b_i + M(1 - v_i), \quad \forall i \in \mathcal{I}, \tag{8b}$$

$$\|\boldsymbol{\epsilon}\|_{\ell} \leq \tau, \tag{8c}$$

$$v_i \in \{0, 1\}, \quad \forall i \in \mathcal{I}. \tag{8d}$$

Let $(\mathbf{v}^k, \boldsymbol{\epsilon}^k)$ be a feasible solution to (8) with respect to $\hat{\mathbf{x}}^k$ and $\mathcal{A}_k = \{i \in \mathcal{I} \mid v_i^k = 1\}$ be a subset of $\mathcal{I}$ for each $k \in \mathcal{K}$, and construct the set of subsets $\mathscr{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_K\}$. Finding exactly $\lceil \theta K \rceil$ subsets in $\mathscr{A}$ such that their intersection has maximum cardinality is equivalent to solving MSI with $\kappa = \lceil \theta K \rceil$. Note that, for each $k \in \mathcal{K}$, there can be multiple solutions for $\mathbf{v}^k$ satisfying (8), each leading to a different $\mathcal{A}_k$. As a result, $\mathscr{A}$ may not be unique. Thus, $\mathbf{MQIO}(\mathcal{K}, \tau, \theta)$ is equivalent to solving the MSI problem multiple times with different $\mathscr{A}$s and finding the maximum cardinality. Therefore, $\mathbf{MQIO}(\mathcal{K}, \tau, \theta)$ is at least as hard as MSI. $\square$

Dawande et al. (2001) show that a general MSI problem can be reformulated as a version of the biclique problem (called the maximum one-sided edge cardinality problem).

4.2. Biclique Representation of the Solution

Motivated by its connection to the MSI problem, we also cast the MQIO problem as a biclique problem with a bipartite graph constructed as follows. We first create a node k in V_1 for each data point $\hat{\mathbf{x}}^k$ and a node i in V_2 for each $\mathbf{a}^i$. With every node $k \in V_1$ and $i \in V_2$, we associate an edge $e = (k, i) \in \mathbb{E}$ if $v_i^k = 1$, where $\mathbf{v}^k$ denotes a feasible solution to Problem (8) with respect to $\hat{\mathbf{x}}^k$. Then, we define a biadjacency matrix $\bar{\mathbf{D}} \in \{0, 1\}^{K \times m}$ associated with graph $G = (V_1 \cup V_2, \mathbb{E})$ as follows.

$$\bar{\mathbf{D}}_{ki} = \begin{cases} 1, & \text{if } e = (k, i) \in \mathbb{E}, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Matrix $\bar{\mathbf{D}}$ can be built by solving Problem (8) for each $k \in \mathcal{K}$ to find a feasible solution $\mathbf{v}^k$ and letting the kth row of $\bar{\mathbf{D}}$ be $\mathbf{v}^{k'}$. The following result shows how an inverse-feasible solution can be found from the $\bar{\mathbf{D}}$ matrix.

Proposition 8. *If there exists $\bar{\mathbf{D}}$ that satisfies (9) and has an all-one submatrix whose rows and columns correspond to $\bar{\mathcal{S}}$ and $\bar{\mathcal{A}}$, respectively, where $\bar{\mathcal{S}} \subseteq \mathcal{K}$, $|\bar{\mathcal{S}}| \geq \theta K$, and $\bar{\mathcal{A}} \subseteq \mathcal{I}$, then there exists a solution $(\bar{\mathbf{c}}, \{\bar{\boldsymbol{\epsilon}}^k\}_{k\in\mathcal{K}}, \bar{\mathbf{y}}, \bar{\mathcal{S}})$ feasible for $\mathbf{QIO}(\mathcal{K}, \tau, \theta)$, where $\bar{\mathbf{c}} \in \text{cone}(\{\mathbf{a}^i\}_{i\in\bar{\mathcal{A}}})$.*

Proposition 8 suggests that, if we can find a matrix $\bar{\mathbf{D}}$ that has an all-one submatrix with at least $\lceil \theta K \rceil$ rows, there exists a corresponding inverse-feasible cost vector for the QIO (hence, MQIO) problem. The following example further illustrates this.

Example 4. Consider Example 3 again (see Figure 2(a)) and the QIO model with the following parameters: $\tau = 1$, $\theta = 0.8$, $\|\cdot\|_{\ell} = \|\cdot\|_{\infty}$, and $\|\cdot\|_p = \|\cdot\|_{\infty}$. Let $(\bar{\mathbf{v}}^k, \bar{\boldsymbol{\epsilon}}^k)$ be a feasible solution to Problem (8) for each data point k: we have $\bar{\mathbf{v}}^k = [1, 1, 0, 0]'$ for $k = 1, 2, 3, 4$, with $\bar{\boldsymbol{\epsilon}}^1 = \begin{bmatrix} -0.5 \\ -0.2 \end{bmatrix}$, $\bar{\boldsymbol{\epsilon}}^2 = \begin{bmatrix} -0.3 \\ -0.2 \end{bmatrix}$, $\bar{\boldsymbol{\epsilon}}^3 = \begin{bmatrix} -0.3 \\ -0.5 \end{bmatrix}$, and $\bar{\boldsymbol{\epsilon}}^4 = \begin{bmatrix} -0.5 \\ -0.5 \end{bmatrix}$, respectively, and $\bar{\mathbf{v}}^5 = [0, 1, 1, 0]'$ with $\bar{\boldsymbol{\epsilon}}^5 = \begin{bmatrix} -0.3 \\ 0.3 \end{bmatrix}$. Then, we can construct a bipartite graph, shown in Figure 2(b), in which each edge identifies (i, k) such that $\bar{v}_i^k = 1$ and the corresponding $\bar{\mathbf{D}}$ matrix in Figure 2(c). Note that this $\bar{\mathbf{D}}$ has an all-one submatrix with rows and columns corresponding to $\bar{\mathcal{S}} = \{1, 2, 3, 4\}$ and $\bar{\mathcal{A}} = \{1, 2\}$, respectively (the shaded rectangle), and $|\bar{\mathcal{S}}| = 4 \geq \theta K$. Now, consider a cost vector $\bar{\mathbf{c}} = 0.5\mathbf{a}^1 + 0.5\mathbf{a}^2 = \begin{bmatrix} -0.5 \\ -0.5 \end{bmatrix} \in \text{cone}(\{\mathbf{a}^i\}_{i\in\bar{\mathcal{A}}})$. We can see that $\bar{\mathbf{c}}$ is inverse-feasible because $(\bar{\mathbf{c}}, \{\bar{\boldsymbol{\epsilon}}^1, \ldots, \bar{\boldsymbol{\epsilon}}^5\}, \bar{\mathbf{y}}, \bar{\mathcal{S}})$, where $\bar{\mathbf{y}} = [0.5, 0.5, 0, 0]'$ is feasible for $\mathbf{QIO}(\mathcal{K}, \tau, \theta)$.

Furthermore, Proposition 8 implies that the number of columns in the all-one submatrix corresponds to the objective function value of $\mathbf{MQIO}(\mathcal{K}, \tau, \theta)$, which is to be maximized. Thus, it is important to find a submatrix

Figure 2. Illustrative Example for the Construction of $\bar{\mathbf{D}}$ and Its All-One Submatrix
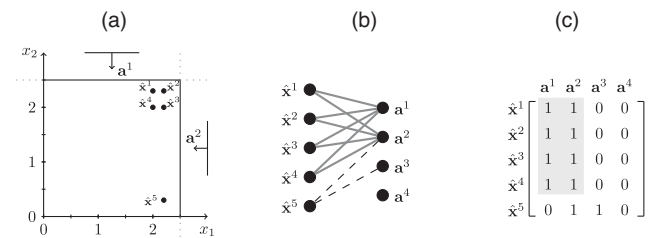
Notes. (a) Forward problem geometry. (b) Bipartite graph. (c) Matrix $\bar{\mathbf{D}}$ and all-one submatrix.

with as many columns as possible. Building on these observations, in the next section, we propose specific algorithms to find an all-one submatrix of $\bar{\mathbf{D}}$ with at least $\lceil \theta K \rceil$ rows and as many columns as possible.

4.3. Algorithms

Although matrix $\bar{\mathbf{D}}$ is constructed using $\{\mathbf{v}^k\}_{k \in \mathcal{K}}$ obtained by solving Problem (8) for each data point k, because there may be multiple solutions for (8), there may be multiple $\bar{\mathbf{D}}$ matrices achievable. Thus, it is important to find a $\bar{\mathbf{D}}$ matrix that can lead to an all-one submatrix of a desirable size as described, which eventually leads to a desirable feasible solution to **MQIO**$(\mathcal{K}, \tau, \theta)$. In this section, we now present our main algorithms: we first present an exact algorithm that solves modifications of (8) repeatedly until a $\bar{\mathbf{D}}$ matrix leading to an optimal solution to the MQIO problem is found, followed by two heuristic variants in which modifications of (8) or their relaxations are solved only K times to more efficiently construct a $\bar{\mathbf{D}}$ matrix that leads to a reasonably good feasible solution to the MQIO problem.

4.3.1. Exact Algorithm.
Our first idea is that we solve Problem (8) with a weighted objective function for each k such that resulting solutions $\{\mathbf{v}^k\}_{k \in \mathcal{K}}$ are similar to each other, thus leading to similar rows of $\bar{\mathbf{D}}$. For each data point k, consider the following modification to (8):

$$\underset{\mathbf{v}, \boldsymbol{\epsilon}}{\text{maximize}} \quad \sum_{i \in \mathcal{I}} w_i^k v_i \tag{10a}$$

$$\text{subject to} \quad b_i \le \mathbf{a}^{i\prime}(\hat{\mathbf{x}}^k - \boldsymbol{\epsilon}) \le b_i + M(1 - v_i), \quad \forall i \in \mathcal{I}, \tag{10b}$$

$$\|\boldsymbol{\epsilon}\|_\ell \le \tau, \tag{10c}$$

$$v_i \in \{0, 1\}, \quad \forall i \in \mathcal{I}. \tag{10d}$$

For the first data point, that is, $k = 1$, we solve Problem (10) with respect to $\hat{\mathbf{x}}^1$, where $w_i^k = 1$ for all $i \in \mathcal{I}$ (i.e., equal initial weights), which finds a solution $\mathbf{v}^1$ with as many as ones as possible. For $k \ge 2$, we solve Problem (10) with respect to $\hat{\mathbf{x}}^k$ with the updated weight parameter $\mathbf{w}^k = \boldsymbol{\vartheta}^k / \|\boldsymbol{\vartheta}^k\|$ and $\vartheta_i^k = \sum_{\ell=1}^{(k-1)} v_i^\ell$; that is, we assign a higher weight to column i for which v_i^ℓ was assigned one more often for the previous data points $\ell = 1, \ldots, k-1$. The resulting vector $\mathbf{v}^{k\prime}$ is then inserted into the kth row of $\bar{\mathbf{D}}$. Note that, because the weights are updated in a sequential manner, the order of the data points can impact the resulting $\bar{\mathbf{D}}$. We assume without loss of generality that the data points are sorted in the increasing order of the sum of the distances to the rest of the data set (i.e., $\sum_{k \in \mathcal{K}} \|\hat{\mathbf{x}}^1 - \hat{\mathbf{x}}^k\|_\ell \le \ldots \le \sum_{k \in \mathcal{K}} \|\hat{\mathbf{x}}^K - \hat{\mathbf{x}}^k\|_\ell$); we find this ordering particularly effective because starting with data points that

are closer to the rest of the data set leads to more similar $\mathbf{v}^k$s in subsequent iterations.

Next, because the construction of $\bar{\mathbf{D}}$ depends on the solution $\mathbf{v}^1$ for Problem (10) with respect to the first data point $\hat{\mathbf{x}}^1$ and this problem may have multiple solutions, it is important to consider different solutions for this problem to place different "seeds" for $\bar{\mathbf{D}}$ (i.e., different first rows of $\bar{\mathbf{D}}$), which allows exploring different inverse solutions. To do so, once an optimal solution to Problem (10) with respect to $\hat{\mathbf{x}}^1$ is found, we impose a cut to exclude the solution from the feasible region and resolve the problem repeatedly.

Finally, once a certain $\bar{\mathbf{D}}$ matrix is found, its all-one submatrix (i.e., a biclique in the bipartite graph corresponding to $\bar{\mathbf{D}}$) with at least $\lceil \theta K \rceil$ rows and as many columns as possible can be found (if it exists) by solving the associated maximum clique problem. Although any existing algorithm for a general maximum clique problem would work, we use an MIP formulation to solve the problem exactly, which we call Clique$(\bar{\mathbf{D}}, \theta)$; this formulation can be found in Appendix B.1.

This algorithm is formally presented in Algorithm 2. In lines 4–9, Problem (10) is solved with updated weights and cuts. To generate different first rows of $\bar{\mathbf{D}}$ (i.e., different $\mathbf{v}^1$'s), a cut defined in line 11 is added to the problem in line 5. The following result shows that, under a certain condition, this algorithm finds inverse-feasible cost vectors that correspond to the optimal solution to **MQIO**$(\mathcal{K}, \tau, \theta)$, hence, exact. We call this algorithm $\bar{\mathbf{D}}$-Alg-Exact.

Proposition 9. *Let* $(\mathbf{v}^*, \mathbf{u}^*, \{\boldsymbol{\epsilon}^{k*}\}_{k \in \mathcal{K}})$ *be an optimal solution to* **MQIO**$(\mathcal{K}, \tau, \theta)$ *and* $z^* = \sum_{i \in \mathcal{I}} v_i^*$. *Let* $\mathcal{V}_z = \{\mathbf{v}^1 \in \{0, 1\}^m \mid \exists (\mathbf{v}^1, \boldsymbol{\epsilon}^1)$ *satisfying* (10b)–(10d) *with respect to* $\hat{\mathbf{x}}^1$, $\mathbf{v}^{1\prime}\mathbf{e} = z\}$. *If* $u_1^* = 1$, *then Algorithm 2 returns* $z^{Alg} = z^*$ *and* $\mathbf{v}^{Alg} = \mathbf{v}^*$ *in* $r \le \sum_{z=z^*}^n |\mathcal{V}_z|$ *iterations.*

Once the algorithm is run, the condition $u_1^* = 1$ can be easily checked by whether the first row of $\bar{\mathbf{D}}_r$ is also included in its all-one submatrix, where r is the iteration at which the most recent $\mathbf{v}^{Alg}$ was achieved. Our numerical results show that the condition is met in all instances for which the algorithm was run completely (not interrupted by time limit). If this condition is not met, although the algorithm may still generate a reasonably good solution, the user may reorder the data points to place a different data point in the position of $k = 1$ and rerun the algorithm. Note that the number of times Problem (10) for $\hat{\mathbf{x}}^1$ with cuts added should be solved in Algorithm 2 can grow exponentially in the size of the set $\mathcal{I}$ in (10) as it depends on the number of feasible solutions for $\mathbf{v}^1$; however, our numerical results show that Algorithm 2 still outperforms solving the MQIO problem as a single large-scale MIP.

Algorithm 2 ($\bar{\mathbf{D}}$-Alg-Exact)

    Result: $z^{Alg}, \mathbf{v}^{Alg}, \mathcal{C}^{Alg}$

    Input: $\mathbf{A}, \mathbf{b}, \hat{\mathcal{X}}, \tau, \theta$

1. $r \leftarrow 1, \mathcal{G}_1 \leftarrow \emptyset, z^{Alg} \leftarrow -\infty$
2.   **while** $|\mathcal{G}_r| \geq z^{Alg}$ **do**
3. $\mathbf{w}^1 \leftarrow \mathbf{e}, \bar{\mathbf{D}}_r \leftarrow [\ ]$
4.   **for** $k = 1, \dots, K$ **do**
5.   Find $\begin{cases} \mathbf{v}^k \in \arg\max_{\mathbf{v}}\{\mathbf{w}^{k'}\mathbf{v} \mid (10b)\text{–}(10d)\}, \\ \qquad \text{if } r = 1 \\ \mathbf{v}^k \in \arg\max_{\mathbf{v}}\{\mathbf{w}^{k'}\mathbf{v} \mid (10b)\text{–}(10d), \\ \qquad \sum_{i \in \mathcal{G}_{r'}} v_i \leq |\mathcal{G}_{r'}| - 1, \ \forall r' = 2, \dots, r\}, \\ \qquad \text{otherwise} \end{cases}$

         with $\hat{\mathbf{x}}^k$ as input data
6.   Insert $\mathbf{v}^{k'}$ to the kth row of $\bar{\mathbf{D}}_r$
7. $\vartheta_i^k \leftarrow \sum_{\ell=1}^k v_i^\ell$ for each $i \in \mathcal{I}$
8.   Update weights $\mathbf{w}^{k+1} \leftarrow \boldsymbol{\vartheta}^k / \|\boldsymbol{\vartheta}^k\|$
9.   **end**
10.   Solve Clique($\bar{\mathbf{D}}_r, \theta$) to find the all-one submatrix of $\bar{\mathbf{D}}_r$. Define $\bar{\mathbf{v}}^r \in \{0,1\}^m$ where ones correspond to columns of the submatrix.
11. $\mathcal{G}_{r+1} \leftarrow \{i \mid v_i^1 = 1 \text{ (i.e., } (1,i) \text{ entry of } \bar{\mathbf{D}}_r \text{ is 1)}\}$
12.   **if** $\bar{\mathbf{v}}^{r'}\mathbf{e} > z^{Alg}$ **then**
13. $z^{Alg} \leftarrow \bar{\mathbf{v}}^{r'}\mathbf{e}$
14. $\mathbf{v}^{Alg} \leftarrow \bar{\mathbf{v}}^r$
15.   **end**
16. $r \leftarrow r + 1$
17.   **end**
18.   Construct the set of inverse-feasible cost vectors $\mathcal{C}^{Alg} = \text{cone}(\{\mathbf{a}^i\}_{i:v_i^{Alg}=1})$

4.3.2. Heuristics. The exact algorithm (Algorithm 2) can be computationally burdensome as it solves Problem (10) for $\hat{\mathbf{x}}^1$ repeatedly with cuts added. To reduce the burden, we propose Algorithm 2 without the use of cuts as a heuristic, that is, finding only one $\bar{\mathbf{D}}$ and terminating Algorithm 2 after the first iteration. This algorithm is faster than the exact algorithm as it solves Problem (10) only K times, albeit not exactly; we call this algorithm $\bar{\mathbf{D}}$-Alg-Heuristic.

We also propose an even faster heuristic in Appendix B.2, which solves relaxations of (10) to more efficiently construct an alternative biadjacency matrix (i.e., instead of $\bar{\mathbf{D}}$). In this relaxed problem (RP), $\mathbf{v}$ is relaxed to be a continuous variable (i.e., $\mathbf{v} \in [0,1]^m$), and weighted ℓ_1 minimization is employed to make $\mathbf{1} - \mathbf{v}$ sparse. We call this algorithm $\bar{\mathbf{D}}^{RP}$-Alg-Heuristic, indicating that the underlying biadjacency matrix of the algorithm is constructed via the RP formulation (see Appendix B.2 for more details).

The proposed algorithm framework can lead to many different variants depending on how the weights are updated and how cuts are generated. For example, the algorithms can be modified to accommodate a huge data set by applying the sequential weight updating only to the first few MIPs (say $\bar{K}$ of them), which leads to the weight parameters $\mathbf{w}^{\bar{K}}$, and solving the remaining $(K - \bar{K})$ MIPs in parallel with the same weights $\mathbf{w}^{\bar{K}}$. Also, the algorithms can be modified to infer cost vectors in an online manner based on data sets that become available through different time points and $(\mathbf{A}, \mathbf{b})$ parameters changing over time; Appendix B.3 shows more details about this algorithm.

5. Numerical Results

We examine the performance of the algorithms proposed for the MQIO model using various sized LP instances. We also assess forward and inverse stability of the MQIO model as well as the previous inverse LP method using the measures proposed in Section 2. Finally, we demonstrate the stability performance of the MQIO model in the diet recommendation and transshipment applications.

5.1. Performance of the Algorithms

We evaluate the performance of the proposed algorithms for randomly generated LP instances with $n \in \{15, 50\}$, $m \in \{100, 300\}$, $K \in \{35, 200, 500\}$, and $\tau \in \{3, 3.5, 4\}$. An instance is defined by a tuple (n, m, K, τ), and each instance was solved twice with $\theta = 0.75$ and 0.85. For each instance, as a preprocessing step, we excluded $\mathbf{a}^i$'s that were not inverse-feasible a priori from the construction of $\mathbf{c}$ (see Algorithm 1). Table 1 shows the results for the MQIO problem (7) obtained by the Gurobi solver (Gurobi Optimization, LLC 2020) (labeled "MIP"), $\bar{\mathbf{D}}$-Alg-Exact, $\bar{\mathbf{D}}$-Alg-Heuristic, and $\bar{\mathbf{D}}^{RP}$-Alg-Heuristic, averaged over two results with $\theta = 0.75$ and 0.85. Recall that the objective function value represents the number of basis vectors ($\mathbf{a}^i$'s) to construct the set of inverse-feasible cost vectors. All optimization problems were solved by Gurobi 7.5 with a four-core 3.6-GHz processor and 32 GB memory.

For small instances in which $(n, m) = (15, 100)$, the exact MIP formulation found an optimal solution within the time limit of 5,000 seconds for most cases. Note that the solution time for MIP decreases as τ goes up; we conjecture that this is because the problem becomes easier as the threshold distance τ becomes more "generous." The exact algorithm achieved the same objective values as the MIP values within significantly less time. Both heuristics found objective values reasonably close to the exact values and were even faster. The computational benefit of the proposed algorithms becomes more clear for larger instances; the MIP solver failed to solve any instance with $(n, m) = (50, 300)$. As expected, there is a clear trade-off between the algorithms. In all instances, $\bar{\mathbf{D}}$-Alg-Exact returned the greatest objective values but was the slowest, whereas

Table 1. MQIO Results via MIP Solver, $\bar{\mathbf{D}}$-Alg-Exact, $\bar{\mathbf{D}}$-Alg-Heuristic, and $\bar{\mathbf{D}}^{RP}$-Alg-Heuristic

		Objective function value				Time, s			
(n, m, K)	τ	MIP	$\bar{\mathbf{D}}$-Exact	$\bar{\mathbf{D}}$-Heur	$\bar{\mathbf{D}}^{RP}$-Heur	MIP	$\bar{\mathbf{D}}$-Exact	$\bar{\mathbf{D}}$-Heur	$\bar{\mathbf{D}}^{RP}$-Heur
(15, 100, 35)	3.0	14.00[a]	14.50	11.50	11.00	5,000.00	12.42	1.32	0.18
	3.5	15.00	15.00	13.00	10.50	2,441.31	26.61	0.72	0.20
	4.0	15.00	15.00	12.00	9.50	28.62	5.77	0.60	0.20
(15, 100, 200)	3.0	12.50[a]	12.50	11.00	11.50	5,000.00	103.03	2.77	0.90
	3.5	15.00	15.00	14.00	10.50	766.59	77.94	2.35	1.05
	4.0	15.00	15.00	15.00	13.00	426.08	2.18	2.18	0.88
(50, 300, 35)	3.0	26.50[a]	37.50	30.50	25.50	5,000.00	2,000.15	43.26	0.77
	3.5	28.00[a]	44.50	36.50	27.00	5,000.00	3,511.62	34.28	0.83
	4.0	24.50[a]	49.00	41.00	28.50	5,000.00	2,181.47	26.31	0.85
(50, 300, 200)	3.0	0.00[a]	33.00	27.00	20.00	5,000.00	1,393.53	145.27	4.23
	3.5	0.00[a]	40.50	29.00	26.00	5,000.00	2,811.66	182.90	4.75
	4.0	0.00[a]	47.50	33.00	27.00	5,000.00	2,705.12	140.94	5.33
(50, 300, 500)	3.0		27.00	27.00	10.50	5,000.00	384.28	384.28	10.23
	3.5		35.00	29.50	13.50	5,000.00	775.12	375.80	12.21
	4.0		43.50	37.00	15.50	5,000.00	3,027.96	186.59	12.84

Note. Empty cells indicate instances for which no feasible solution was found within time limit.

[a]Instances not solved to optimality within the time limit (5,000 seconds).

$\bar{\mathbf{D}}^{RP}$-Alg-Heuristic was the fastest algorithm, yet the solution quality was not as good as the other two algorithms. Although the results suggest that both heuristics can be used for very large instances, because $\bar{\mathbf{D}}^{RP}$-Alg-Heuristic solves relaxations for each $\hat{\mathbf{x}}^k$ instead of MIPs, we suggest using this algorithm for instances with very large n and m.

5.2. Stability of MQIO

5.2.1. Forward Stability. To assess the forward stability of our proposed MQIO model, we created an instance with $(n, m, K, \tau) = (15, 100, 35, 3)$ and $\theta = 0.75$, for which we know the true optimal objective value is $\sum_{i \in \mathcal{I}} v_i^* = 15$. To see the effect of the number of $\mathbf{a}^i$'s used to form an inverse solution (i.e., $\sum_{i \in \mathcal{I}} v_i$) on forward stability, we solved the inverse problem on this instance repeatedly, each with a constraint $\sum_{i \in \mathcal{I}} v_i \leq h$, $h \in \{1, 2, \ldots, 15\}$. Each problem was solved exactly via the MQIO formulation. Note that the case in which $h = 1$ can be considered similar to the previous inverse LP model (2) in that it finds a single cost vector identical to one of the $\mathbf{a}^i$'s. For each h, once the set of inverse-feasible cost vectors, $\mathcal{C}_h$, was found, we randomly generated 50 cost vectors $\mathbf{c} \in \mathcal{C}_h$ and computed $d(\hat{\mathcal{X}}, \mathbf{x}^*(\mathbf{c})) = \max_{k \in \mathcal{S}_h} \{\|\hat{\mathbf{x}}^k - \mathbf{x}^*(\mathbf{c})\|_\infty\}$, where $\mathcal{S}_h$ denotes the set of chosen data points in iteration h and $\mathbf{x}^*(\mathbf{c}) \in$ argmin $\mathbf{FO}(\mathbf{c})$ as an estimate for the forward stability measure.

Figure 3(a) shows the resulting distances for each $h \in \{1, 2, \ldots, 15\}$: each box plot represents 50 values of $d(\hat{\mathcal{X}}, \mathbf{x}^*(\mathbf{c}))$ from 50 different cost vectors. We observe that, when h is low (i.e., few $\mathbf{a}^i$'s are used for generating a $\mathbf{c}$ vector), the distance d is significantly greater than τ, hence, forward-unstable. In other words, if this cost vector is used for the forward problem, the resulting solutions may not be close to the input data. On the other hand, as h increases (i.e., as more $\mathbf{a}^i$'s are used) the cost vector produces a solution that is closer to the data, eventually within the threshold distance τ (indicated by the horizontal dashed line). The overall decreasing trend of the distance d supports the idea of maximizing $\sum_{i \in \mathcal{I}} v_i$ as a surrogate objective function to maximize forward stability.

5.2.2. Inverse Stability. To evaluate the inverse stability of MQIO, we investigated the effect of θ and τ on how sensitive the inverse solution set is to changes in the data set $\hat{\mathcal{X}}$. Again, an instance with $(n, m, K) = (15, 100, 35)$ was considered. First, we considered the MQIO model with different τ values increasing from two to three by 0.1 (the objective value of MQIO increases monotonically as τ increases). For each τ value, we first solved the MQIO model with the given data set $\hat{\mathcal{X}}$ to find the initial inverse-feasible set $\hat{\mathcal{C}}$. We then shifted $\lfloor (1 - \theta)K \rfloor + 1$ data points in $\hat{\mathcal{X}}$: $\{\hat{\mathbf{x}}^k + \gamma^k\}_{k \in \mathcal{K}}$, where γ^k is uniformly distributed in $[0, \Gamma]^n$ if $\hat{\mathbf{x}}^k$ is chosen to be shifted, and $\gamma^k = \mathbf{0}$ otherwise. Given this shifted data, denoted by $\tilde{\mathcal{X}}(\Gamma)$, we resolved the MQIO problem to find the new inverse-feasible set $\tilde{\mathcal{C}}$. We repeated this process while increasing Γ until $\tilde{\mathcal{C}}$ had no common cost vector shared by the initial set $\hat{\mathcal{C}}$, that is, $\hat{\mathcal{C}} \cap \tilde{\mathcal{C}} = \emptyset$. When such a Γ value was reached, we computed the distance $d(\hat{\mathcal{X}}, \tilde{\mathcal{X}}(\Gamma)) = \sum_{k \in \mathcal{K}} \|\hat{\mathbf{x}}^k - \tilde{\mathbf{x}}^k(\Gamma)\|_\infty$ as an estimate for the inverse stability measure in Section 2.

Figure 3(b) shows the result of this experiment for all τ values and $\theta = 0.75, 0.85$, and 1. For a fixed value of θ, the distance $d(\hat{\mathcal{X}}, \tilde{\mathcal{X}}(\Gamma))$ increases as τ increases (i.e., more $\mathbf{a}^i$ vectors are included in the inverse-feasible set), which means the model becomes more tolerant to changes in the data set and, hence, more

Figure 3. (Color online) Forward and Inverse Stability Performance of MQIO
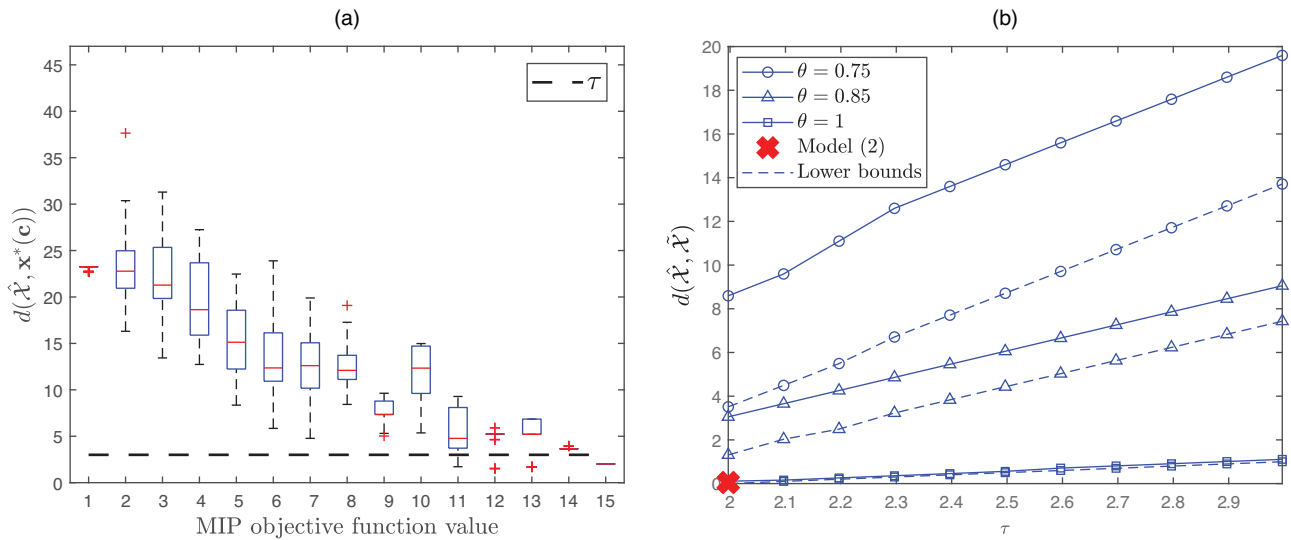
Notes. (a) Distances between the data set and solutions achieved by inverse-feasible cost vectors. (b) Data shift distance until no initial inverse-feasible solution remains feasible.

inverse-stable. For a fixed value of τ, Figure 3(b) also shows that increasing θ decreases inverse stability, reinforcing the idea that allowing for no or few outliers can lead to an "impatient" inverse model that can easily switch its solutions affected by such data imperfection.

Note that the estimated inverse stability values in Figure 3(b) are, in fact, upper bounds on the true inverse stability values. To show this, we refer the reader to Formulation (A.3) in Appendix A.3, whose optimal value is the true inverse stability value for the MQIO model. Because no initial cost vector in $\hat{\mathcal{C}}$ remains inverse-feasible given the shifted data $\tilde{\mathcal{X}}$, $\tilde{\mathcal{X}}$ is a feasible solution to Formulation (A.3) with the objective value of $d(\hat{\mathcal{X}}, \tilde{\mathcal{X}})$. Because (A.3) is a minimization problem, its optimal value (i.e., the true inverse stability value) cannot be greater than $d(\hat{\mathcal{X}}, \tilde{\mathcal{X}})$. Figure 3(b) shows that the lower bounds on the inverse stability measure (based on Proposition 3) are nontrivial, and because the estimated values are upper bounds, the difference between the lower bounds and the true inverse stability values must be even less than shown in Figure 3(b).

Finally, we conducted a similar experiment to assess the inverse stability of the previous inverse model (2). We solved Model (2) with $\hat{\mathcal{X}}$ to find an initial inverse solution $\hat{\mathbf{c}}$, resolved the problem repeatedly with $\tilde{\mathcal{X}}(\Gamma)$ with increasing Γ values until the resulting cost vector was different from $\hat{\mathbf{c}}$, and computed the distance between the original data set and the shifted one, which is shown as a thick $\times$ marker in Figure 3(b). The result is almost identical to the lowest inverse stability achieved by the MQIO model (i.e., with $\theta = 1$ and smallest τ).

5.3. Diet Recommendation

Literature on diet prediction/recommendation using historical data largely focuses on a "direct" replication of diets for which the goal is to create diets that are closest to the old diets in terms of some distance function (e.g., ℓ_2 distance between new and old diet vectors) (Darmon et al. 2006, Perignon et al. 2016). However, such an approach may not work if there is any change in the constraints of the underlying diet problem, such as changes in nutritional requirements or available foods. In this case, learning the objective function that represents one's preferences instead and creating new diets "indirectly" is a more robust and transferable way of learning the individual's diet behavior (similar motivations can be found in learning driving behaviors in Abbeel and Ng (2004)). Moreover, by doing so, the model can find an *optimal* diet with respect to the inferred preferences, whereas replicating the diets directly can inherit undesirable diet patterns from the past.

We use the database from the National Health and Nutrition Examination Surveys, which includes nutritional requirements and nutrition facts per serving for each food type, and build a diet problem for a subset of the foods and nutrients from the database to keep the experiment simple. We classify the foods into food "types" (see Table D.1 in Appendix D.1). We assume that, once the number of servings for each food type is determined, more detailed decisions (e.g., specific menus) can be made by dietitians based on this. We believe this is a realistic consideration as there exist myriad different menus. We first solve the diet problem with some arbitrary cost vector, assumed to reflect true preferences of an individual, and find $\mathbf{x}^*$ that represents the number

Figure 4. (Color online) Comparison of Diets Recommended by the MQIO Model and the Previous Inverse Model
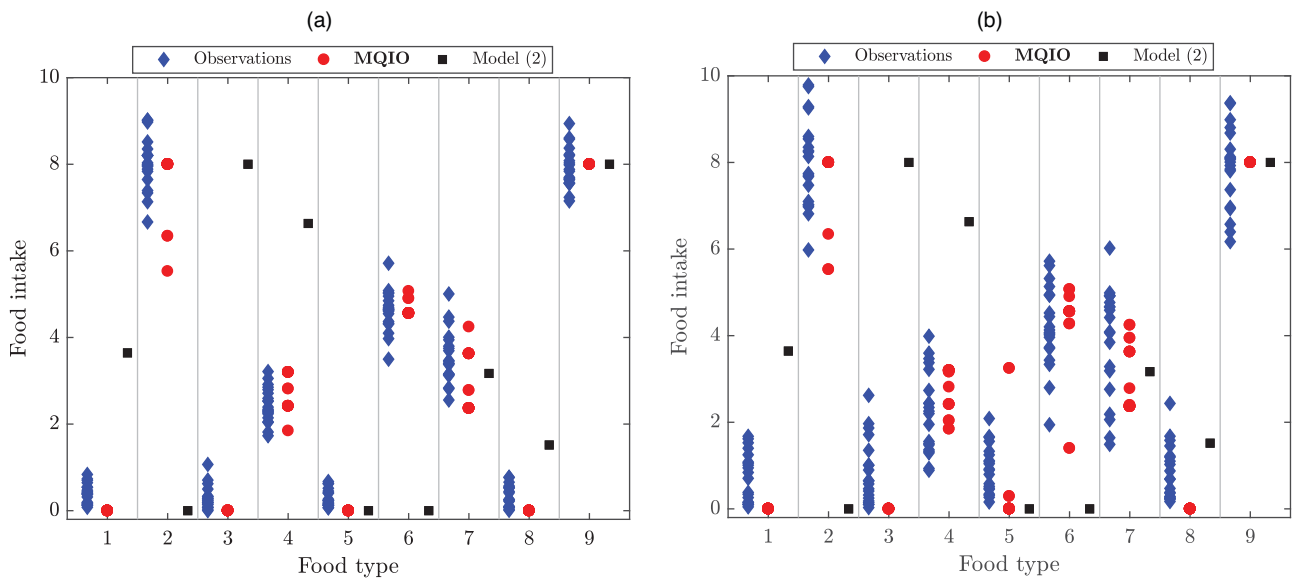
Notes. (a) $\sigma = 0.2$. (b) $\sigma = 1.0$.

of servings for food type i per day. In practice, if the data come from some unknown system, the assumed forward LP feasible region may not "fit" the data well. In this case, the user needs to tweak some parameters of the forward LP (e.g., constraint parameters or the list of variables), similar to linear regression.

Given x^* with no noise, both the MQIO and previous models can find the true cost vector and generate the same x^* as a forward solution. However, as one's diet behavior is assumed inconsistent and noisy, to examine the impact of data noise on both models, we generate multiple diets $x^* + \gamma$, where γ is a normal-distributed random noise vector $\gamma \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$, where $\mathbf{I}$ is the identity matrix of an appropriate dimension, which form the input data set $\hat{\mathcal{X}}$.

We apply the MQIO model as well as the previous inverse model (2) with the same data set $\hat{\mathcal{X}}$ as input and compare the resulting recommended diets. For MQIO, we find a set of inverse-feasible cost vectors $\mathcal{C}^*$ and solve the forward diet problem 20 times each with a different $\mathbf{c}$ vector selected from $\mathcal{C}^*$ to generate a set of diet recommendations. Because the previous inverse model finds a single cost vector, we solve the diet problem with this vector only once. Figure 4(a) shows diet recommendations obtained by the MQIO model (circular markers) are much closer to the input data (diamond-shaped markers; $\sigma = 0.2$ to create the noisy data set $\hat{\mathcal{X}}$) than those from the previous inverse model are (squared markers), reinforcing the improved stability in the MQIO model. Also note that the MQIO model leads to multiple such diet recommendations based on the multiple cost vectors.

In Figure 4(b), we further increase the noise in the input data by increasing σ from 0.2 to 1. Although the

previous model generates the same recommendation, the MQIO model "adapts to" the increased variability in the observations and generates more diverse recommendations for some food types; for example, food types 5–7. In summary, the MQIO model generates recommendations that are consistent with the individual's past behavior and stable in the face of data noise; thus, the objective function found by the model can better predict one's eating behavior.

Finally, we show that the MQIO framework is amenable to the case in which data points are collected at different time points and the goal of the inverse problem is to infer cost vectors based on more recent data points. Let $\hat{x}^t$ be a data point collected at time $t = 1, \ldots, T$. We let the threshold parameter τ_t vary over time: the more recent $\hat{x}^t$ is, the smaller τ_t we impose (i.e., the closer we want to make this data point to optimality). Figure 5 shows the result of such a modified model with time-stamped diet observations. Darker diamond markers represent more recent observations. As more recent data were collected, the modified MQIO model generates recommendations (circular markers) that are closer to the more recent ones. This result suggests that our MQIO framework can offer modeling flexibility that can lend itself to adaptive settings in which the preference function can be updated over time.

5.4. Transshipment Problem

We demonstrate the use of our inverse model for online learning in the transshipment application, in which production and shipment costs are inferred adaptively as batches of decisions are observed sequentially over time. We consider a transshipment

Figure 5. (Color online) Results of the MQIO Model for Diet Behavior Changing over Time
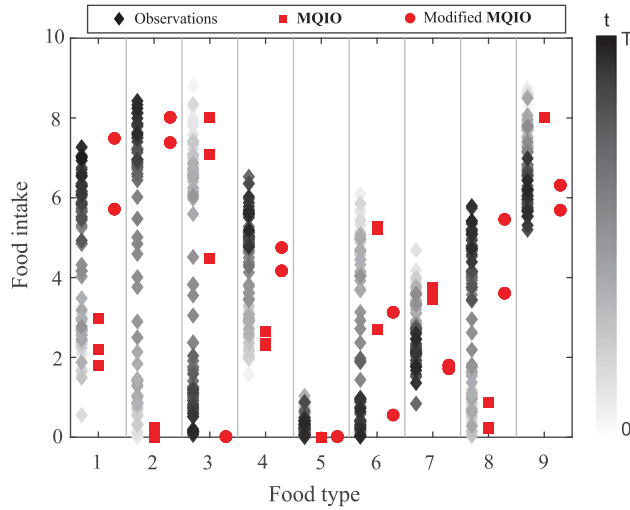
problem with one distribution node, two supply nodes, and two demand nodes (see Appendix D.2). Suppose, at each time t, demand $\mathbf{d}^t$ is revealed, and the DM makes a set of decisions on production $(\mathbf{x}^{(p)})$ and transshipment $(\mathbf{x}^{(t)})$ to minimize the total cost $\mathbf{c}^{(p)\prime}\mathbf{x}^{(p)} + \mathbf{c}^{(t)\prime}\mathbf{x}^{(t)}$, where $\mathbf{c}^{(p)}$ and $\mathbf{c}^{(t)}$ denote production and transshipment costs, respectively. Let $\mathbf{c} = [\mathbf{c}^{(p)}; \mathbf{c}^{(t)}]$ and $\mathbf{x} = [\mathbf{x}^{(p)}; \mathbf{x}^{(t)}]$. Let $\hat{\mathcal{X}}_t$ denote the set of decisions observed at time t. We assume that there is a true cost vector $\mathbf{c}^{true}$ (shown in Appendix D.2) and generate a data set $\hat{\mathcal{X}}_t$ by adding noise to an optimal solution with respect to $\mathbf{c}^{true}$ and randomly generated demand $\mathbf{d}^t \sim U[0, 1.1]$.

We use an online learning extension to $\bar{\mathbf{D}}$-Alg-Heuristic (see Appendix B.3) to infer the set of cost vectors $\mathcal{C}_t$ at each time t from the data sets collected through time t. We consider $T = 600$ (i.e., 600 sequential batches of data) and $\theta = 0.75$. At each iteration $t = 1, \ldots, 600$, we randomly select 20 cost vectors $\mathbf{c}^t \in \mathcal{C}_t$ and solve the corresponding forward problem with demand $\mathbf{d}^t$ to obtain a forward solution $\bar{\mathbf{x}}^t$. We assess the performance of the algorithm at each time t in terms of the distance between our forward solution $\bar{\mathbf{x}}^t$ and data batch $\hat{\mathcal{X}}_t$, that is, $d(\hat{\mathcal{X}}_t, \bar{\mathbf{x}}^t) = \max_{\mathbf{x} \in \hat{\mathcal{X}}_t} \|\mathbf{x} - \bar{\mathbf{x}}^t\|_\infty$.

Figure 6(a) shows the average distance $\sum_{t=1}^{\bar{t}} d(\hat{\mathcal{X}}_t, \bar{\mathbf{x}}^t)/\bar{t}$ at iteration $\bar{t}$ over 20 trials, that is, 20 randomly selected cost vectors. Each light-color line shows each distance per trial, and the dark line shows the average. As $\bar{t}$ grows, the average distance between the data set and the forward solution converges. In fact, the inferred set of cost vectors remains unchanged from iteration 415 onward, and importantly, this set includes $\mathbf{c}^{true}$. That is, as more data are collected, the algorithm successfully finds a set of cost vectors that represent the preferences encoded in the true cost vector and generate forward solutions close to the given data (hence, forward-stable). Additional results on the comparison between the objective function values achieved by our forward solution $\bar{\mathbf{x}}^t$ and those from the given data are available in Appendix D.2.

We also test the algorithm with different values of $\theta \in \{0.75, 0.85, 1\}$. Figure 6(b) shows that, as θ decreases, the algorithm takes more iterations to converge, which is intuitive as fewer data points are considered relevant at each iteration, requiring the algorithm to collect more data until it finally arrives at a certain set of cost vectors. When $\theta = 1$, although the

Figure 6. (Color online) Results of the Online Learning Extension for the Transshipment Problem
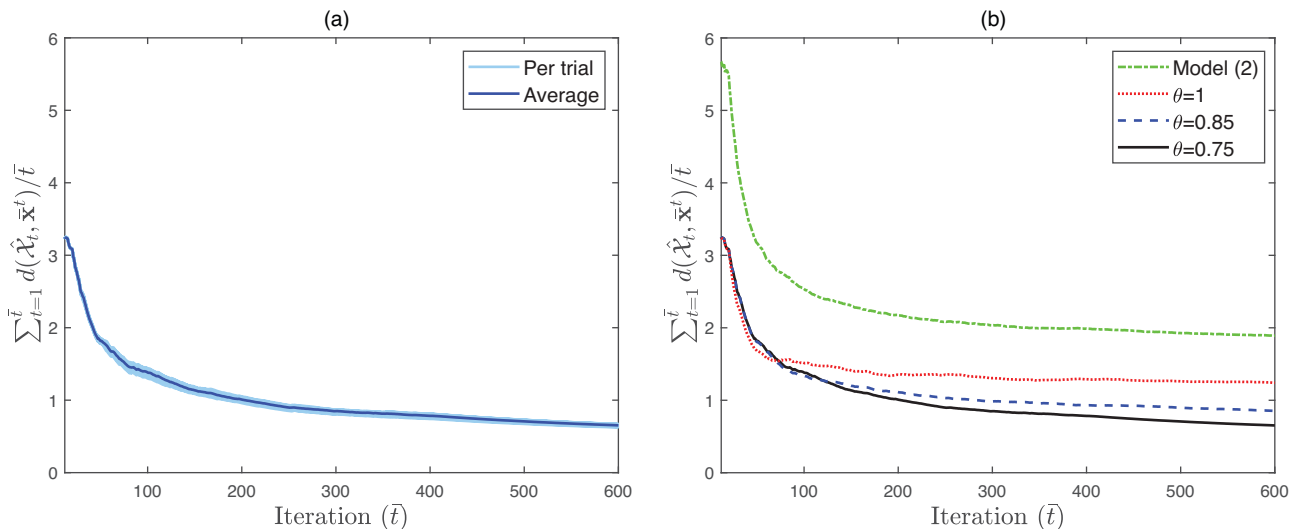
Notes. (a) Distance between inversely optimized forward solution and the data set. (b) Performance of the algorithm with different θ values and Model (2).

distance converges more quickly, the resulting inverse set does not include the true cost vector because the algorithm with $\theta = 1$ is more susceptible to data noise. This again supports the need to explicitly account for potential outliers in inverse LP. Finally, Figure 6(b) also shows the performance of the previous inverse model (2) in this setting: at each iteration $\bar{t}$, we find a cost vector by solving (2) with all data points up to time $\bar{t}$. This model fails to find the true cost vector at all iterations. Also, the distance between its corresponding forward solution and the data is greater than that from the MQIO model.

The performance of the online learning algorithm (i.e., Algorithm B.2) also depends on the variability of demand parameters. When the interval of the demand distribution increases, the true vector is found more slowly. We conjecture that this is because increased variability in demands causes the feasible region of the forward problem to vary more, rendering some constraints redundant, and thus, the algorithm cannot detect some of the $\mathbf{a}^i$'s needed to construct the true cost vector.

6. Conclusion

In this paper, we develop a new inverse LP method that can capture noise, errors, and uncertainty in the input data and infer cost vectors that are more stable than those obtained by the previous methods. We formulate the model as a large-scale MIP and develop efficient algorithms by exploiting its connection to the well-known biclique problems. Our inverse method is demonstrated in the diet recommendation and transshipment applications in which past data used as input can be noisy and inaccurate. Many directions for future research exist. For example, preferences of a large group of DMs can be clustered into smaller representative groups using the quantile statistic-based inverse optimization approach. A goodness-of-fit measure that is amenable to the quantile inverse model can be useful for assessing the validity of the assumed forward LP, which remains our future work. Extending the inverse nonlinear programming techniques to accommodate the quantile-based framework is also an interesting future work, which can allow to assess our proposed inverse model in a broader context.

Appendix A. Supplemental Materials for the Inverse Model

A.1. Finding the Threshold Parameter τ

The threshold parameter τ for the QIO model (i.e., (5)) can be set by adding a reasonable margin to the minimum possible value that keeps the model feasible, which can be found by solving the following problem in which τ is now a variable:

$$\underset{\tau, \mathbf{c}, \{\epsilon^k\}_{k \in \mathcal{K}}, \mathbf{y}, \mathcal{S}}{\text{minimize}} \{\tau \mid \text{(5b)–(5i)}\}. \qquad (A.1)$$

Problem (A.1) finds a cost vector such that the θ quantile optimality error is minimized; this is similar to the structure of the least quantile method in the regression context in Bertsimas and Mazumder (2014). The following result shows the solution structure of an optimal cost vector for the preceding problem (proof can be found in Appendix C).

Proposition A.1. *If Problem (A.1) is feasible, then there exists an optimal $\mathbf{c}$ for (A.1) such that $\mathbf{c} = \mathbf{a}^i$ for some $i \in \mathcal{I}$.*

Proposition A.1 suggests that the search for an optimal cost vector for (A.1) can be done by evaluating each hyperplane $i = 1, \ldots, m$, that is, solving Formulation (C.4) for each i and each k (see proof of Proposition A.1 in Appendix C) and finding i that induces the minimum τ^*; Formulation (C.4) is a convex optimization problem and, thus, is straightforward to solve.

To further illustrate the implication of Proposition A.1, consider the forward feasible region and the initial data points in Example 1 (Figure 1(a)) and Problem (A.1) with $\theta = 0.5$ and $\|\cdot\|_\ell = \|\cdot\|_\infty$. Cost vector $\mathbf{c} = \mathbf{a}^1$ is the optimal solution to Problem (A.1), leading to the objective value of $\tau^* = 0.2$ with the optimality errors $\|\epsilon^1\|_\infty = \|\epsilon^2\|_\infty = 0.2$; the objective function values for (A.1) achieved by $\mathbf{c} = \mathbf{a}^2$, $\mathbf{c} = \mathbf{a}^3$, and $\mathbf{c} = \mathbf{a}^4$ are 0.3, 2, and 2, respectively.

A.2. Finding Big M Parameters

For each $k \in \mathcal{K}$, consider the following formulation that finds $\mathbf{x} \in \mathcal{X}$ that has maximum ℓ_∞ distance from $\hat{\mathbf{x}}^k$:

$$\underset{\mathbf{x}}{\text{maximize}} \{\|\mathbf{x} - \hat{\mathbf{x}}^k\|_\infty \mid \mathbf{x} \in \mathcal{X}\}. \qquad (A.2)$$

Let d_k denote the optimal value of (A.2) with respect to $\hat{\mathbf{x}}^k$. Given that $\mathcal{X}$ is bounded, d_k is also bounded. To find d_k, we first reformulate (A.2) as $\max_{j \in \mathcal{J}} \max_{\mathbf{x} \in \mathcal{X}} \{|x_j - \hat{x}_j^k|\}$; for each $j \in \mathcal{J}$, if $\bar{\mathbf{x}}$ denotes an optimal solution to the inner max problem, then $\max\{|\bar{x}_j - \hat{x}_j^k|\} = \max\{\bar{x}_j - \hat{x}_j^k, \hat{x}_j^k - \bar{x}_j\}$. As a result, we have $d_k = \max_{j \in \mathcal{J}} \max\{\max_{\mathbf{x} \in \mathcal{X}} \{x_j - \hat{x}_j^k\}, \max_{\mathbf{x} \in \mathcal{X}} \{\hat{x}_j^k - x_j\}\}$, that is, d_k can be obtained by solving $2n$ LPs and selecting the largest objective value. Note that $M_2 = \max_{k \in \mathcal{K}} \{d_k\}$ is a sufficiently large number valid for Constraint (6c). More efficiently, one may use a constraint-specific parameter $M_{2k} = d_k$ for each $k \in \mathcal{K}$.

To find M_1, we use Cauchy–Schwarz and Minkowski inequalities as follows. For each $i \in \mathcal{I}$ and $k \in \mathcal{K}$, by the Cauchy–Schwarz inequality, $\mathbf{a}^{i'}(\hat{\mathbf{x}}^k - \epsilon^k) \leq \|\mathbf{a}^i\|_2 (\|\hat{\mathbf{x}}^k - \epsilon^k\|_2)$. Using the Minkowski inequality, we have $\|\hat{\mathbf{x}}^k - \epsilon^k\|_2 \leq \|\hat{\mathbf{x}}^k\|_2 + \|\epsilon^k\|_2$ for each $k \in \mathcal{K}$. Moreover, for each vector $\epsilon^k \in \mathbb{R}^n$, we have $\|\epsilon^k\|_2 = \sqrt{\sum_{i=1}^n |\epsilon_i^k|^2} \leq \sqrt{\sum_{i=1}^n (\max_i\{|\epsilon_i^k|\})^2} = \sqrt{n \times (\max_i\{|\epsilon_i^k|\})^2} = \sqrt{n}\|\epsilon^k\|_\infty$. It follows that $\mathbf{a}^{i'}(\hat{\mathbf{x}}^k - \epsilon^k) - b_i \leq \|\mathbf{a}^i\|_2 (\|\hat{\mathbf{x}}^k - \epsilon^k\|_2) - b_i \leq \|\mathbf{a}^i\|_2 (\|\hat{\mathbf{x}}^k\|_2 + \|\epsilon^k\|_2) - b_i \leq \|\mathbf{a}^i\|_2 (\|\hat{\mathbf{x}}^k\|_2 + \sqrt{n}\|\epsilon^k\|_\infty) - b_i$. From the previous paragraph, we have $\|\epsilon^k\|_\infty \leq d_k$ for each $k \in \mathcal{K}$. Hence, choosing $M_1 = \max_{i \in \mathcal{I}} \{\|\mathbf{a}^i\|_2 (\max_{k \in \mathcal{K}} \{\|\hat{\mathbf{x}}^k\|_2\} + \sqrt{n} \max_{k \in \mathcal{K}} \{d_k\}) + \max_{i \in \mathcal{I}} \{-b_i\}$ is valid for each constraint of (6d). As a more efficient and constraint-specific alternative, one may choose $M_{1ik} = \|\mathbf{a}^i\|_2 (\|\hat{\mathbf{x}}^k\|_2 + \sqrt{n}d_k) - b_i$ for constraint (6d) for each $i \in \mathcal{I}$ and $k \in \mathcal{K}$.

The M parameter in Formulation (8), with respect to each $k \in \mathcal{K}$, can be calculated similarly to M_1 with d_k

replaced by τ, that is, $M = \max_{i \in \mathcal{I}}\{\|\mathbf{a}^i\|_2\}(\|\hat{\mathbf{x}}^k\|_2 + \tau\sqrt{n}) + \max_{i \in \mathcal{I}}\{-b_i\}$.

A.3. Computing the Inverse Stability Measure

Recall that inverse stability measure (3) is defined as the minimum distance of data shift such that all inverse-feasible cost vectors are no longer feasible for the inverse model with respect to the shifted data. Given a data set $\hat{\mathcal{X}}$, let $\hat{\mathcal{C}}$ be the set of inverse-feasible cost vectors for $QIO(\hat{\mathcal{X}}, \tau, \theta)$. As Theorem 1 suggests, every inverse-feasible cost vector $\mathbf{c} \in \hat{\mathcal{C}}$ is a conic combination of some $\mathbf{a}^i$'s that are also inverse-feasible. The following result shows that, to check whether all inverse-feasible cost vectors in $\hat{\mathcal{C}}$ are no longer feasible under the shifted data, it is sufficient to check whether all inverse-feasible $\mathbf{a}^i$'s whose conic hull forms $\hat{\mathcal{C}}$ are no longer feasible.

Lemma A.1. *Let $\hat{\mathcal{C}}$ be the set of inverse-feasible cost vectors for $QIO(\hat{\mathcal{X}}, \tau, \theta)$ and $\hat{\mathcal{I}} = \{i \in \mathcal{I} \mid \mathbf{a}^i \in \hat{\mathcal{C}}\}$. Let $\tilde{\mathcal{X}}$ denote the shifted data set and $\tilde{\mathcal{C}}$ the set of inverse-feasible cost vectors for $QIO(\tilde{\mathcal{X}}, \tau, \theta)$. If no $\mathbf{a}^i, i \in \hat{\mathcal{I}}$ is inverse-feasible for $QIO(\tilde{\mathcal{X}}, \tau, \theta)$, then no $\mathbf{c} \in \hat{\mathcal{C}}$ is inverse-feasible for $QIO(\tilde{\mathcal{X}}, \tau, \theta)$.*

Lemma A.1 suggests that, to compute the inverse stability measure for $\hat{\mathcal{C}}$ given the original data set $\hat{\mathcal{X}}$, all we need is to find a minimally shifted set of the data points, $\tilde{\mathcal{X}}$ such that no $\mathbf{a}^i, i \in \hat{\mathcal{I}}$ is inverse-feasible for $QIO(\tilde{\mathcal{X}}, \tau, \theta)$. Given $\hat{\mathcal{I}}$, to make $\mathbf{c} = \mathbf{a}^i$ infeasible for $QIO(\tilde{\mathcal{X}}, \tau, \theta)$ for all $i \in \hat{\mathcal{I}}$, we need to ensure that there are more than $\lfloor(1-\theta)K\rfloor$ data points in $\tilde{\mathcal{X}}$ whose minimum distance from each $\mathcal{X}_i, i \in \hat{\mathcal{I}}$, is strictly greater than τ. This observation leads to the following formulation.

$$\underset{\{\mathbf{x}^k\}_{k \in \mathcal{K}}, \{\mathcal{S}^i\}_{i \in \hat{\mathcal{I}}}}{\text{minimize}} \quad d(\hat{\mathcal{X}}, \mathcal{X}) = \sum_{k \in \mathcal{K}}\|\mathbf{x}^k - \hat{\mathbf{x}}^k\|_\ell \tag{A.3a}$$

$$\text{subject to} \quad D^i(\mathbf{x}^k) = \min_{\mathbf{z} \in \mathcal{X}_i}\{\|\mathbf{x}^k - \mathbf{z}\|_\ell\}, \quad \forall i \in \hat{\mathcal{I}}, k \in \mathcal{K}, \tag{A.3b}$$

$$D^i(\mathbf{x}^k) \geq \tau + \sigma, \quad \forall k \in \mathcal{S}^i, i \in \hat{\mathcal{I}}, \tag{A.3c}$$

$$|\mathcal{S}^i| \geq \lfloor(1-\theta)K\rfloor + 1, i \in \hat{\mathcal{I}}. \tag{A.3d}$$

In this formulation, variable $\mathbf{x}^k$ is considered a shifted data point from $\hat{\mathbf{x}}^k$. In Constraint (A.3b), $D^i(\mathbf{x}^k)$ computes the minimum distance between some shifted data point $\mathbf{x}^k$ and $\mathcal{X}_i$, which is then ensured to be no less than $\tau + \sigma$ for at least $\lfloor(1-\theta)K\rfloor + 1$ data points by Constraints (A.3c) and (A.3d), where σ denotes an infinitesimal positive constant. The objective function minimizes the sum of distances between the original data points and the shifted ones. As computing the exact inverse stability value would require $D^i(\mathbf{x}^k) > \tau$ in place of Constraint (A.3c), the optimal value of the problem is only infinitesimally greater than the exact inverse stability value, and we treat this value as the true inverse stability value (the inverse stability measure is defined on the one-dimensional open set). Note that Problem (A.3) is always feasible as one can let $\mathbf{x}^k, k \in \mathcal{K}$ be arbitrarily far from each $\mathcal{X}_i, i \in \hat{\mathcal{I}}$. Problem (A.3) is nonconvex, and thus, it is still challenging to compute its optimal value exactly. However, we use this formulation to compute the lower bound for the true inverse stability value, which is easy to compute (see Proposition 3 and its proof).

A.4. Upper Bound for the Forward Stability Measure

In this section, we show that an optimal solution to the MQIO model leads to an upper bound on forward stability measure (4). We make the following assumption: for each $i \in \mathcal{I}$, there exists a constant ρ_i such that $\|\mathbf{x}^1 - \mathbf{x}^2\|_\ell \leq \rho_i$ for any $\mathbf{x}^1, \mathbf{x}^2 \in \mathcal{X}_i$ and $\ell \geq 1$. The following result shows that, if such ρ_i values are known a priori, an upper bound for the forward stability measure for a cost vector found by the MQIO model can be computed.

Proposition A.2. *Let $(\bar{\mathbf{v}}, \bar{\mathbf{u}}, \{\bar{\boldsymbol{\epsilon}}\}_{k \in \mathcal{K}})$ be a feasible solution to $MQIO(\mathcal{K}, \tau, \theta)$, $\bar{\mathcal{I}} = \{i \in \mathcal{I} \mid \bar{v}_i = 1\}$, $\bar{\mathcal{S}} = \{k \in \mathcal{K} \mid \bar{u}_k = 1\}$, and $\bar{\mathbf{c}} = \sum_{i \in \bar{\mathcal{I}}}\lambda_i\mathbf{a}^i$, where $\lambda_i > 0$ for all $i \in \bar{\mathcal{I}}$. Then, $\max_{\mathbf{x} \in \mathcal{X}^*(\bar{\mathbf{c}})}\{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\} \leq \min_{\mathbf{x} \in \cap_{i \in \bar{\mathcal{I}}}\mathcal{X}_i}\{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\} + \min_{i \in \bar{\mathcal{I}}}\{\rho_i\}$ for all $k \in \bar{\mathcal{S}}$.*

Because $\max_{\mathbf{x} \in \mathcal{X}^*(\bar{\mathbf{c}})}\{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\}$ is bounded above for each $k \in \bar{\mathcal{S}}$, the forward stability measure given any distance function d in (4), for example, $\max_{k \in \bar{\mathcal{S}}}\max_{\mathbf{x} \in \mathcal{X}^*(\bar{\mathbf{c}})}\{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\}$, is also bounded above.

Given the same inverse-feasible cost vector $\bar{\mathbf{c}}$ in Proposition A.2, if $\ell = \infty$, the exact forward stability measure (4) can be written as $\max_{k \in \bar{\mathcal{S}}}\{\max_{\mathbf{x} \in \mathcal{X}^*(\bar{\mathbf{c}})}\{\|\mathbf{x} - \hat{\mathbf{x}}^k\|_\infty\}\}$. The optimal value of the inner max problem for each k can be obtained by solving $2n$ LPs: $z_{kj}^{(1)} = \max_{\mathbf{x} \in \mathcal{X}^*(\bar{\mathbf{c}})}\{x_j - \hat{x}_j^k\}$ and $z_{kj}^{(2)} = \max_{\mathbf{x} \in \mathcal{X}^*(\bar{\mathbf{c}})}\{\hat{x}_j^k - x_j\}$ for each $j \in \mathcal{J} = \{1, \ldots, n\}$ and selecting the largest value among them, that is, $\max_{j \in \mathcal{J}}\{z_{kj}^{(1)}, z_{kj}^{(2)}\}$. Thus, we have the exact forward stability measure: $\max_{k \in \bar{\mathcal{S}}}\{\max_{\mathbf{x} \in \mathcal{X}^*(\bar{\mathbf{c}})}\{\|\mathbf{x} - \hat{\mathbf{x}}^k\|_\infty\}\} = \max_{k \in \bar{\mathcal{S}}}\max_{j \in \mathcal{J}}\{z_{kj}^{(1)}, z_{kj}^{(2)}\}$.

A.5. Extensions to the MQIO model

In Section 3, we assume the constraint parameters $(\mathbf{A}, \mathbf{b})$ remain the same over different observations. However, the proposed formulations can also apply when $\mathbf{b}$ varies over k (i.e., $\mathbf{b}^k$ for observation k) as they only exploit the structure of $\mathbf{A}$. Should $\mathbf{A}$ vary over k (i.e., $\mathbf{A}^k$ for observation k), an inverse-feasible cost vector can be obtained by replacing Constraint (6b) by $b_i^k \leq \mathbf{a}^{ki\prime}(\hat{\mathbf{x}}^k - \boldsymbol{\epsilon}^k) \leq b_i^k + M_1(1 - v_i^k)$, $\forall i \in \mathcal{I}, \forall k \in \mathcal{K}$, constructing observation-specific conic hulls $\mathcal{C}_k$ using $\mathbf{v}^k$ similarly as in Section 3, and finding the intersection of the multiple conic hulls (e.g., via the following additional constraint: $\mathbf{c} = \sum_{i \in \mathcal{I}}\lambda_i^k\mathbf{a}^{ki}$ and $0 \leq \lambda^k \leq \mathbf{v}^k$ for all $k \in \mathcal{S}$).

Additionally, we note that the MQIO model is also more stable than the previous inverse LP method when the parameters $(\mathbf{A}, \mathbf{b})$ are subject to uncertainty (i.e., the realized parameters may turn out to be different from the assumed ones). Suppose that both the previous inverse LP and our MQIO models are used to infer cost vectors given some "nominal" parameters $(\bar{\mathbf{A}}, \bar{\mathbf{b}})$. Because the previous model finds a cost vector that is identical to one of the $\mathbf{a}^i$'s, a slight change in the corresponding $\mathbf{a}^i$ can make the inferred cost vector no longer inverse-feasible. On the other hand, if we let $\bar{\mathcal{C}}$ be the set of inverse-feasible cost vectors from the original MQIO model, there exists $\mathbf{c} \in \bar{\mathcal{C}}$ that is still inverse-feasible for the *realized* MQIO problem (with $(\tilde{\mathbf{A}}, \tilde{\mathbf{b}}) \neq (\bar{\mathbf{A}}, \bar{\mathbf{b}})$) as long as there exists $\mathbf{x}^k$ within τ-distance from at least $\lceil\theta K\rceil$ data points such that $\mathbf{x}^k \in \cap_{i \in \tilde{\mathcal{I}}}\mathcal{X}_i$ for some set $\tilde{\mathcal{I}} \subseteq \mathcal{I}$ and $\bar{\mathcal{C}} \cap \text{cone}(\{\mathbf{a}^i\}_{i \in \tilde{\mathcal{I}}}) \neq \emptyset$. In general, if $(\mathbf{A}, \mathbf{b})$ is subject to uncertainty, the MQIO problem lends itself to traditional robust optimization or stochastic programming techniques as it is a linear MIP.

Finally, should the cost vector be constrained, say $\mathbf{D}\mathbf{c} \le \mathbf{d}$, the MQIO formulation can be rewritten as follows:

$$\operatorname*{maximize}_{\mathbf{v}, \mathbf{u}, \mathbf{c}, \lambda, \{\epsilon^k\}_{k \in \mathcal{K}}} \left\{ \sum_{i \in \mathcal{I}} v_i \,\middle|\, \mathbf{D}\mathbf{c} \le \mathbf{d}, \mathbf{c} = \sum_{i \in \mathcal{I}} \lambda_i \mathbf{a}^i, 0 \le \lambda \le \mathbf{v}, \text{(6b)–(6e)} \right\},$$
(A.4)

where the first three constraints are added to ensure that the resulting $\mathbf{c}$ satisfies the cost vector constraint and is still a conic combination of some $\mathbf{a}^i$'s.

Appendix B. Supplemental Materials for the Algorithms

B.1. The Biclique Problem Formulation

Let $\bar{\mathbf{D}} \in \{0,1\}^{K \times m}$ be the biadjacency matrix defined as (9). The following MIP finds (if it exists) an all-one submatrix of $\bar{\mathbf{D}}$ (i.e., a biclique) with at least $\lceil \theta K \rceil$ rows (i.e., data points) and as many columns ($\mathbf{a}^i$'s) as possible.

$$\text{Clique}(\bar{\mathbf{D}}, \theta): \quad \operatorname*{maximize}_{\mathbf{v}^{clq}, \mathbf{u}^{clq}} \quad \sum_{i \in \mathcal{I}} v_i^{clq} \tag{B.1a}$$

$$\text{subject to} \quad u_k^{clq} + v_i^{clq} \le 1, \quad \forall (k,i) \in \mathcal{K} \times \mathcal{I} \text{ such that } \bar{\mathbf{D}}_{ki} = 0, \tag{B.1b}$$

$$\sum_{k \in \mathcal{K}} u_k^{clq} \ge \theta K, \tag{B.1c}$$

$$v_i^{clq}, u_k^{clq} \in \{0,1\}, \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}. \tag{B.1d}$$

In (B.1), u_k^{clq} and v_i^{clq} are one if node $k \in V_1$ and $i \in V_2$ are selected to be in the clique, respectively, and zero otherwise. The objective function maximizes the number of nodes selected from V_2 (i.e., columns of the all-one submatrix of $\bar{\mathbf{D}}$). Constraint (B.1b) ensures that at most one of the nodes can be selected from $k \in V_1$ and $i \in V_2$ if there is no edge corresponding to (k, i), and Constraint (B.1c) guarantees that at least $\lceil \theta K \rceil$ nodes from V_1 (i.e., rows of the submatrix) are selected. Once Problem (B.1) is solved, the submatrix can be retrieved by selecting rows and columns of $\bar{\mathbf{D}}$ corresponding to $\{k \in \mathcal{K} \mid u_k^{clq} = 1\}$ and $\{i \in \mathcal{I} \mid v_i^{clq} = 1\}$, respectively.

B.2. Alternative Heuristic

The algorithms proposed in Section 4.3 can still be computationally burdensome because the construction of each $\bar{\mathbf{D}}$ requires solving the MIP problem (10) K times. To address this, we propose a relaxation of Problem (10) using the idea of weighted ℓ_1 minimization (Candes et al. 2008). For each $k \in \mathcal{K}$, we solve the following relaxed problem, which has only continuous variables:

$$\operatorname*{minimize}_{\alpha, \epsilon} \quad \sum_{i \in \mathcal{I}} w_i^k \alpha_i$$

$$\text{subject to} \quad b_i \le \mathbf{a}^{i'} (\hat{\mathbf{x}}^k - \epsilon) \le b_i + M\alpha_i, \quad \forall i \in \mathcal{I}, \tag{B.2a}$$

$$\|\epsilon\|_\ell \le \tau, \tag{B.2b}$$

$$\alpha \ge \mathbf{0}. \tag{B.2c}$$

Then, we construct a matrix $\bar{\mathbf{D}}^{RP} \in \{0,1\}^{K \times m}$:

$$\bar{\mathbf{D}}_{ki}^{RP} = \begin{cases} 1, & \text{if } \alpha_i^k = 0, \\ 0, & \text{otherwise}, \end{cases} \tag{B.3}$$

where α^k is a feasible solution to (B.2) with respect to $\hat{\mathbf{x}}^k$. Note that the construction of $\bar{\mathbf{D}}^{RP}$ is more efficient than that of $\bar{\mathbf{D}}$. The following result shows the relationship between a solution to Problem (B.2) and a solution to Problem (10).

Lemma B.1. *Given an observation $\hat{\mathbf{x}}^k$, let $(\bar{\alpha}^k, \bar{\epsilon}^k)$ be a feasible solution to Problem (B.2) and $\bar{\mathcal{I}} = \{i \in \mathcal{I} \mid \bar{\alpha}_i^k = 0\}$. Then, there exists a feasible solution $(\bar{\mathbf{v}}^k, \bar{\epsilon}^k)$ to Problem (10) with respect to $\hat{\mathbf{x}}^k$, where $\bar{v}_i^k = 1$ for $i \in \bar{\mathcal{I}}$ and $\bar{v}_i^k = 0$ for $i \in \mathcal{I} \setminus \bar{\mathcal{I}}$.*

Next, similar to Proposition 8, the following result shows how a feasible solution to the QIO (hence, MQIO) model can be retrieved from $\bar{\mathbf{D}}^{RP}$.

Proposition B.1. *If there exists $\bar{\mathbf{D}}^{RP}$ that satisfies (B.3) and has an all-one submatrix whose rows and columns correspond to $\bar{\mathcal{S}}$ and $\bar{\mathcal{A}}$, respectively, where $\bar{\mathcal{S}} \subseteq \mathcal{K}$, $|\bar{\mathcal{S}}| \ge \theta K$, and $\bar{\mathcal{A}} \subseteq \mathcal{I}$, then there exists a solution $(\bar{\mathbf{c}}, \{\bar{\epsilon}^k\}_{k \in \mathcal{K}}, \bar{\mathbf{y}}, \bar{\mathcal{S}})$ feasible for $\mathbf{QIO}(\mathcal{K}, \tau, \theta)$ such that $\bar{\mathbf{c}} \in \text{cone}(\{\mathbf{a}^i\}_{i \in \bar{\mathcal{A}}})$.*

We use the same instance provided in Example 4 to illustrate the result of Proposition B.1.

Example B.1. Let $(\bar{\alpha}^k, \bar{\epsilon}^k)$ denote the optimal solution to Problem (B.2) associated with each data point $k = 1, \dots, 5$: we have $\bar{\alpha}^k = [0, 0, \bar{\alpha}_3^k > 0, \bar{\alpha}_4^k > 0]'$ for $k = 1, 2, 3, 4$, (with $\bar{\epsilon}^1 = \begin{bmatrix} -0.5 \\ -0.2 \end{bmatrix}$, $\bar{\epsilon}^2 = \begin{bmatrix} -0.3 \\ -0.2 \end{bmatrix}$, $\bar{\epsilon}^3 = \begin{bmatrix} -0.3 \\ -0.5 \end{bmatrix}$, and $\bar{\epsilon}^4 = \begin{bmatrix} -0.5 \\ -0.5 \end{bmatrix}$), and $\bar{\alpha}^5 = [\bar{\alpha}_1^5 > 0, 0, 0, \bar{\alpha}_4^5 > 0]'$ (with $\bar{\epsilon}^5 = \begin{bmatrix} -0.3 \\ 0.3 \end{bmatrix}$). We then construct a bipartite graph by defining node i for $\mathbf{a}^i$, node k for $\hat{\mathbf{x}}^k$, and edge (k, i) where $\bar{\alpha}_i^k = 0$, which leads to the same bipartite graph in Figure 2(b). As a result, the corresponding biadjacency matrix $\bar{\mathbf{D}}^{RP}$ is identical to the $\bar{\mathbf{D}}$ matrix in Figure 2(c). Similar to Example 4, we can obtain an inverse-feasible cost vector from the all-one submatrix of $\bar{\mathbf{D}}^{RP}$.

Similar to the algorithms in Section 4.3, there can be multiple $\bar{\mathbf{D}}^{RP}$s because there may be multiple optimal solutions to Problem (B.2). We propose a heuristic, which we call $\bar{\mathbf{D}}^{RP}$-Alg-Heuristic (indicating that it is a variant of $\bar{\mathbf{D}}$-Alg-Heuristic via the relaxed Problem (B.2)) via a similar weighting approach to find a matrix $\bar{\mathbf{D}}^{RP}$ and the same clique problem to find the all-one submatrix of $\bar{\mathbf{D}}^{RP}$ (i.e., Clique($\bar{\mathbf{D}}^{RP}, \theta$)). The pseudo-code for this heuristic can be found in Algorithm B.1.

Algorithm B.1 ($\bar{\mathbf{D}}^{RP}$-Alg-Heuristic)

> **Result**: $\mathcal{C}^*$
> **Input**: $\mathbf{A}, \mathbf{b}, \hat{\mathcal{X}}, \tau, \theta$
1. $\mathbf{w}^1 \leftarrow \mathbf{e}, \bar{\mathbf{D}}^{RP} \leftarrow [\,]$
2. **for** $k = 1, \dots, K$ **do**
3.   Find $\alpha^k \in \operatorname{argmin}_\alpha \{\mathbf{w}^{k'} \alpha \mid \text{(16a)–(16c)}\}$ with $\hat{\mathbf{x}}^k$ as input data
4. $\bar{\mathbf{D}}_{ki}^{RP} \leftarrow 1$ if $\alpha_i^k = 0$; $\bar{\mathbf{D}}_{ki}^{RP} \leftarrow 0$ otherwise // $\bar{\mathbf{D}}_{ki}^{RP} := (k,i)$ entry of $\bar{\mathbf{D}}^{RP}$
5. $\vartheta_i^k \leftarrow \sum_{\ell=1}^k \bar{\mathbf{D}}_{\ell i}^{RP}$ for each $i \in \mathcal{I}$
6.   Update weights $\mathbf{w}^{k+1} \leftarrow \vartheta^k / \|\vartheta^k\|$
7. **end**
8. Solve Clique($\bar{\mathbf{D}}^{RP}, \theta$) to find the all-one submatrix of $\bar{\mathbf{D}}^{RP}$. Define $\mathbf{v}^* \in \{0,1\}^m$ where ones correspond to columns of the submatrix.
9. Construct the set of inverse-feasible cost vectors $\mathcal{C}^* = \text{cone}(\{\mathbf{a}^i\}_{i: v_i^* = 1})$

B.3. Application to Online Learning

The assumption we make in Section 3 is that the data points are available all at once in advance from a fixed forward LP formulation. However, in many application domains, data may become available through different time points in separate batches. Moreover, some parameters of the underlying forward LP formulation may change at each time point as well. In this section, we show how our inverse method can be extended for learning the cost vectors adaptively over time in an online manner.

Suppose, at each time $t \in \mathcal{T} = \{1, \ldots, T\}$, the DM observes an external signal $\mathbf{b} = \mathbf{b}^t$ as the right-hand side of the forward problem and makes a set of decisions $\hat{\mathcal{X}}_t$ with the index set $\mathcal{K}_t$ of cardinality K_t. We assume that the decisions can be noisy. The following formulation is a modification to MQIO that finds inverse-feasible cost vectors for the entire collection of batches $\bigcup_{t \in \mathcal{T}} \hat{\mathcal{X}}_t$:

$$\underset{\mathbf{v}, \mathbf{u}, \{\boldsymbol{\epsilon}^{kt}\}_{k \in \mathcal{K}_t, t \in \mathcal{T}}}{\text{maximize}} \quad \sum_{i \in \mathcal{I}} v_i \tag{B.4a}$$

$$\text{subject to} \quad \mathbf{b}^t \le \mathbf{A}(\hat{\mathbf{x}}^{kt} - \boldsymbol{\epsilon}^{kt}) \le \mathbf{b}^t + M_1(\mathbf{1} - \mathbf{v}),$$
$$\forall k \in \mathcal{K}_t, \ \forall t \in \mathcal{T}, \tag{B.4b}$$

$$\|\boldsymbol{\epsilon}^{kt}\|_\ell \le \tau_t + M_2(1 - u_{kt}), \quad \forall k \in \mathcal{K}_t, \ \forall t \in \mathcal{T}, \tag{B.4c}$$

$$\sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}_t} u_{kt} \ge \theta \sum_{t \in \mathcal{T}} K_t, \tag{B.4d}$$

$$v_i, u_{kt} \in \{0, 1\}, \quad \forall i \in \mathcal{I}, \ \forall k \in \mathcal{K}_t, \ \forall t \in \mathcal{T}. \tag{B.4e}$$

This formulation still aims to solve the inverse problem in the traditional setting, which means it has to wait until all data points are collected, leading to a large-scale MIP. Instead, we propose an online learning approach using a modified version of $\bar{\mathbf{D}}$-Alg-Heuristic in Algorithm B.2.

In this algorithm, we start with an empty $\bar{\mathbf{D}}$ matrix and update it at each time t by finding $\mathbf{v}^* \in \text{argmax}_\mathbf{v} \{\mathbf{w}^{t'} \mathbf{v} \mid (6b)-(6e)\}$ with $\hat{\mathcal{X}}_t$ as input data (where $\mathbf{w}^t$ is an appropriate weight vector at time t) and inserting $\mathbf{v}^{*'}$ into the next empty row of $\bar{\mathbf{D}}$. At each time t, given the updated $\bar{\mathbf{D}}$, we find its largest all-one submatrix and let $\mathbf{v}^t$ be a binary vector with ones corresponding to the columns of the submatrix. Recall that each column of this matrix corresponds to each $\mathbf{a}^i$; thus, $\mathbf{v}^t$ indicates which $\mathbf{a}^i$'s should be used for creating the set of inverse-feasible cost vectors at time t, that is, $\mathcal{C}_t = \text{cone}(\{\mathbf{a}^i\}_{i:v_i^t=1})$. We show in the numerical results section that, with a sufficiently large T, there exists $\mathbf{c} \in \mathcal{C}_T$ that is also inverse-feasible for the MIP (B.4). Again, with such a large T and an excessive number of data points, the MIP problem (B.4) would be computationally extremely challenging, whereas the online learning approach is more efficient as it attempts to solve the problem in a distributed manner.

Algorithm B.2 (Online Algorithm to Find Inverse-Feasible Solutions)

Result: $\mathcal{C}_T$
Input: $\mathbf{A}, \tau, \theta$
1. $\mathbf{w}^1 \leftarrow \mathbf{1}, \bar{\mathbf{D}} \leftarrow [\,]$
2. **for** $t = 1, \ldots, T$ **do**
3. $\quad$ Observe $(\mathbf{b}^t, \hat{\mathcal{X}}_t)$
4. $\quad$ Find $\mathbf{v}^* \in \text{argmax}_\mathbf{v} \{\mathbf{w}^{t'} \mathbf{v} \mid (6b)-(6e)\}$ with $\hat{\mathcal{X}}_t$ as input data and $\mathbf{b} = \mathbf{b}^t$
5. $\quad$ Insert $\mathbf{v}^{*'}$ to the tth row of $\bar{\mathbf{D}}$
6. $\quad$ Solve $\text{Clique}(\bar{\mathbf{D}}, \theta)$ to find the all-one submatrix of $\bar{\mathbf{D}}$. Define $\mathbf{v}^t \in \{0, 1\}^m$ where ones correspond to columns of the submatrix.
7. $\quad$ Construct the set of inverse-feasible cost vectors $\mathcal{C}_t = \text{cone}(\{\mathbf{a}^i\}_{i:v_i^t=1})$
8. $\quad$ Update weights $\mathbf{w}^{t+1} \leftarrow (\mathbf{1} + \mathbf{v}^t)/\|\mathbf{1} + \mathbf{v}^t\|$
9. **end**

Appendix C. Proofs

Proof of Proposition 1. Given $\hat{\mathcal{X}}, \tau, \theta_1$, and θ_2, let $\hat{\mathcal{C}}(\theta_1), \hat{\mathcal{C}}(\theta_2), \tilde{\mathcal{C}}(\theta_1)$, and $\tilde{\mathcal{C}}(\theta_2)$ denote the set of cost vectors obtained by $\mathbf{QIO}(\hat{\mathcal{X}}, \tau, \theta_1), \mathbf{QIO}(\hat{\mathcal{X}}, \tau, \theta_2), \mathbf{QIO}(\tilde{\mathcal{X}}, \tau, \theta_1)$, and $\mathbf{QIO}(\tilde{\mathcal{X}}, \tau, \theta_2)$, respectively, where $\tilde{\mathcal{X}}$ denotes some shifted data set. To show that inverse stability measure (3) for $\hat{\mathcal{C}}(\theta)$ is nondecreasing as θ decreases, we need to show that, given θ_1 and θ_2, if $\theta_2 \le \theta_1$, then $\min_{\tilde{\mathcal{X}}}\{d(\hat{\mathcal{X}}, \tilde{\mathcal{X}}) \mid \hat{\mathcal{C}}(\theta_2) \cap \tilde{\mathcal{C}}(\theta_2) = \emptyset\} \ge \min_{\tilde{\mathcal{X}}}\{d(\hat{\mathcal{X}}, \tilde{\mathcal{X}}) \mid \hat{\mathcal{C}}(\theta_1) \cap \tilde{\mathcal{C}}(\theta_1) = \emptyset\}$. Given $\theta_2 \le \theta_1$, we have $\hat{\mathcal{C}}(\theta_1) \subseteq \hat{\mathcal{C}}(\theta_2)$ and $\tilde{\mathcal{C}}(\theta_1) \subseteq \tilde{\mathcal{C}}(\theta_2)$. Let $\hat{\mathcal{C}}_{\text{diff}} = \hat{\mathcal{C}}(\theta_2) \backslash \hat{\mathcal{C}}(\theta_1)$ and $\tilde{\mathcal{C}}_{\text{diff}} = \tilde{\mathcal{C}}(\theta_2) \backslash \tilde{\mathcal{C}}(\theta_1)$. Then,

$$\min_{\tilde{\mathcal{X}}}\{d(\hat{\mathcal{X}}, \tilde{\mathcal{X}}) \mid \hat{\mathcal{C}}(\theta_2) \cap \tilde{\mathcal{C}}(\theta_2) = \emptyset\}$$
$$= \min_{\tilde{\mathcal{X}}}\{d(\hat{\mathcal{X}}, \tilde{\mathcal{X}}) \mid (\hat{\mathcal{C}}_{\text{diff}} \cup \hat{\mathcal{C}}(\theta_1)) \cap (\tilde{\mathcal{C}}_{\text{diff}} \cup \tilde{\mathcal{C}}(\theta_1)) = \emptyset\}$$
$$= \min_{\tilde{\mathcal{X}}}\{d(\hat{\mathcal{X}}, \tilde{\mathcal{X}}) \mid (\tilde{\mathcal{C}}_{\text{diff}} \cup \tilde{\mathcal{C}}(\theta_1)) \cap (\hat{\mathcal{C}}_{\text{diff}} \cup \hat{\mathcal{C}}(\theta_1)) = \emptyset\}$$
$$= \min_{\tilde{\mathcal{X}}}\{d(\hat{\mathcal{X}}, \tilde{\mathcal{X}}) \mid (\tilde{\mathcal{C}}_{\text{diff}} \cap (\hat{\mathcal{C}}_{\text{diff}} \cup \hat{\mathcal{C}}(\theta_1))) \cup$$
$$\quad (\tilde{\mathcal{C}}(\theta_1) \cap (\hat{\mathcal{C}}_{\text{diff}} \cup \hat{\mathcal{C}}(\theta_1))) = \emptyset\}$$
$$\ge \min_{\tilde{\mathcal{X}}}\{d(\hat{\mathcal{X}}, \tilde{\mathcal{X}}) \mid \tilde{\mathcal{C}}(\theta_1) \cap (\hat{\mathcal{C}}_{\text{diff}} \cup \hat{\mathcal{C}}(\theta_1)) = \emptyset\}$$
$$= \min_{\tilde{\mathcal{X}}}\{d(\hat{\mathcal{X}}, \tilde{\mathcal{X}}) \mid (\tilde{\mathcal{C}}(\theta_1) \cap \hat{\mathcal{C}}_{\text{diff}}) \cup (\tilde{\mathcal{C}}(\theta_1) \cap \hat{\mathcal{C}}(\theta_1)) = \emptyset\}$$
$$\ge \min_{\tilde{\mathcal{X}}}\{d(\hat{\mathcal{X}}, \tilde{\mathcal{X}}) \mid \tilde{\mathcal{C}}(\theta_1) \cap \hat{\mathcal{C}}(\theta_1) = \emptyset\},$$

which completes the proof. $\square$

Proof of Proposition 2. Let $(\bar{\mathbf{c}}, \{\bar{\boldsymbol{\epsilon}}^k\}_{k \in \mathcal{K}}, \bar{\mathbf{y}}, \bar{\mathcal{S}})$ be a feasible solution to $\mathbf{QIO}(\mathcal{K}, \tau, \theta)$. Note that $\bar{y}_i > 0$ for at least one $i \in \mathcal{I}$. Let $\bar{\mathcal{I}} = \{i \in \mathcal{I} \mid \bar{y}_i > 0\}$. From (5b), $\bar{\mathbf{c}} = \sum_{i \in \bar{\mathcal{I}}} \bar{y}_i \mathbf{a}^i$. From (5e), for each $k \in \bar{\mathcal{S}}$, $\sum_{i \in \bar{\mathcal{I}}} \bar{y}_i \mathbf{a}^{i'}(\hat{\mathbf{x}}^k - \bar{\boldsymbol{\epsilon}}^k) = \sum_{i \in \bar{\mathcal{I}}} b_i \bar{y}_i$, and thus, $\sum_{i \in \bar{\mathcal{I}}} \bar{y}_i(\mathbf{a}^{i'}(\hat{\mathbf{x}}^k - \bar{\boldsymbol{\epsilon}}^k) - b_i) = 0$. Because $\bar{y}_i > 0$, we have $\mathbf{a}^{i'}(\hat{\mathbf{x}}^k - \bar{\boldsymbol{\epsilon}}^k) - b_i = 0, \forall i \in \bar{\mathcal{I}}$. Now, pick any arbitrary $\tilde{i} \in \bar{\mathcal{I}}$ and let $\tilde{\mathbf{y}} = \mathbf{e}_{\tilde{i}}$, where $\mathbf{e}_{\tilde{i}}$ denotes the $\tilde{i}$ th unit vector. Then, from (5b), we can construct a new cost vector $\tilde{\mathbf{c}} = \mathbf{a}^{\tilde{i}}$. By replacing $\bar{\mathbf{c}}$ with this $\tilde{\mathbf{c}}$ in (5e), (5e) becomes $\mathbf{a}^{\tilde{i}'}(\hat{\mathbf{x}}^k - \bar{\boldsymbol{\epsilon}}^k) - b_{\tilde{i}} = 0, \forall k \in \bar{\mathcal{S}}$, which still holds because $\tilde{i} \in \bar{\mathcal{I}}$. It is clear that $(\tilde{\mathbf{c}} = \mathbf{a}^{\tilde{i}}, \{\bar{\boldsymbol{\epsilon}}^k\}_{k \in \mathcal{K}}, \tilde{\mathbf{y}} = \mathbf{e}_{\tilde{i}}, \bar{\mathcal{S}})$ also satisfies Constraints (5b)–(5i) and, thus, is feasible for (5). $\square$

Proof of Lemma 1. Given some sets $\bar{\mathcal{A}} \subseteq \mathcal{A}$ and $\bar{\mathcal{S}}$ such that $|\bar{\mathcal{S}}| \ge \theta K$, let $\bar{\mathbf{c}} \in \text{cone}(\{\mathbf{a}^i\}_{i \in \bar{\mathcal{A}}})$ such that $\|\bar{\mathbf{c}}\|_p = 1$. Recall that $\mathcal{X}_i = \{\mathbf{x} \in \mathcal{X} \mid \mathbf{a}^{i'}\mathbf{x} = b_i\}, i \in \mathcal{I}$. We want to show that, if there exists $\bar{\mathbf{x}}^k \in \mathcal{X}_i$ for all $i \in \bar{\mathcal{A}}$ such that $\|\bar{\mathbf{x}}^k - \hat{\mathbf{x}}^k\|_\ell \le \tau$ for all $k \in \bar{\mathcal{S}}$, then there exists $\{\bar{\boldsymbol{\epsilon}}^k\}_{k \in \mathcal{K}}$ and $\bar{\mathbf{y}}$ such that $(\bar{\mathbf{c}}, \{\bar{\boldsymbol{\epsilon}}^k\}_{k \in \mathcal{K}}, \bar{\mathbf{y}}, \bar{\mathcal{S}})$ is feasible for $\mathbf{QIO}(\mathcal{K}, \tau, \theta)$ (i.e., Formulation (5)). Note that $\bar{\mathbf{c}}$ can be written as $\bar{\mathbf{c}} = \sum_{i \in \bar{\mathcal{A}}} \bar{\lambda}_i \mathbf{a}^i$, where

$\bar{\lambda}_i > 0$ for at least one $i \in \bar{\mathcal{A}}$. Let $\bar{\lambda}_i = 0$ for $i \in \mathcal{I} \setminus \bar{\mathcal{A}}$. Because $\bar{\mathbf{x}}^k \in \mathcal{X}^i$, $\forall i \in \bar{\mathcal{A}}$, for each $k \in \bar{\mathcal{S}}$, we have $\mathbf{a}^{i'}\bar{\mathbf{x}}^k = b_i$ for all $i \in \bar{\mathcal{A}}$. Multiplying both sides by $\bar{\lambda}_i$ and summing over all $i \in \bar{\mathcal{A}}$, we have $\sum_{i \in \bar{\mathcal{A}}} \bar{\lambda}_i \mathbf{a}^{i'}\bar{\mathbf{x}}^k = \sum_{i \in \bar{\mathcal{A}}} \bar{\lambda}_i b_i$, and because $\bar{\lambda}_i = 0$ for $i \in \mathcal{I} \setminus \bar{\mathcal{A}}$, we further have $\sum_{i \in \mathcal{I}} \bar{\lambda}_i \mathbf{a}^{i'}\bar{\mathbf{x}}^k = \sum_{i \in \mathcal{I}} \bar{\lambda}_i b_i$; hence, $\bar{\mathbf{c}}'\bar{\mathbf{x}}^k = \mathbf{b}'\bar{\lambda}$ for all $k \in \bar{\mathcal{S}}$. By letting $\bar{\mathbf{y}} = \bar{\lambda}$, we have $\mathbf{A}'\bar{\mathbf{y}} = \bar{\mathbf{c}}$ and $\bar{\mathbf{y}} \geq \mathbf{0}$, which satisfy Constraints (5b) and (5c), respectively. For $k \in \bar{\mathcal{S}}$, let $\bar{\epsilon}^k = \hat{\mathbf{x}}^k - \bar{\mathbf{x}}^k$; for $k \in \mathcal{K} \setminus \bar{\mathcal{S}}$, let $\bar{\epsilon}^k$ be an arbitrary vector in $\mathbb{R}^n$. Then, Constraint (5d) is satisfied because $\hat{\mathbf{x}}^k - \bar{\epsilon}^k = \bar{\mathbf{x}}^k \in \mathcal{X}$, $\forall k \in \bar{\mathcal{S}}$, and Constraint (5e) is satisfied because $\bar{\mathbf{c}}'(\hat{\mathbf{x}}^k - \bar{\epsilon}^k) = \bar{\mathbf{c}}'\bar{\mathbf{x}}^k = \mathbf{b}'\bar{\lambda} = \mathbf{b}'\bar{\mathbf{y}}$, $\forall k \in \bar{\mathcal{S}}$. Constraint (5f) is satisfied because $\|\bar{\epsilon}^k\|_\ell = \|\hat{\mathbf{x}}^k - \bar{\mathbf{x}}^k\|_\ell \leq \tau$, $\forall k \in \bar{\mathcal{S}}$. Finally, (5g)–(5i) are satisfied as it was assumed that $\bar{\mathcal{S}} \subseteq \mathcal{K}$, $|\bar{\mathcal{S}}| \geq \theta K$ and $\|\bar{\mathbf{c}}\|_p = 1$. As a result, $(\bar{\mathbf{c}}, \{\bar{\epsilon}^k\}_{k \in \mathcal{K}}, \bar{\mathbf{y}}, \bar{\mathcal{S}})$ is feasible for $\mathbf{QIO}(\mathcal{K}, \tau, \theta)$. □

Proof of Theorem 1. ($\Rightarrow$) Assume $(\bar{\mathbf{c}} \neq \mathbf{0}, \{\bar{\epsilon}^k\}_{k \in \mathcal{K}}, \bar{\mathbf{y}}, \bar{\mathcal{S}})$ is feasible for $\mathbf{QIO}(\mathcal{K}, \tau, \theta)$, that is, Model (5). Then, from (5b), $\bar{\mathbf{c}} = \sum_{i \in \bar{\mathcal{I}}} \bar{y}_i \mathbf{a}^i$, where $\bar{\mathcal{I}} = \{i \in \mathcal{I} \mid \bar{y}_i > 0\}$. Construct $\bar{\mathbf{v}} \in \mathbb{R}^{|\mathcal{I}|}$ by setting $\bar{v}_i = 1$ for all $i \in \bar{\mathcal{I}}$ and $\bar{v}_i = 0$ for all $i \in \mathcal{I} \setminus \bar{\mathcal{I}}$. Also define $\bar{\mathbf{u}} \in \mathbb{R}^{|\mathcal{K}|}$ such that $\bar{u}_k = 1$ for $k \in \bar{\mathcal{S}}$ and $\bar{u}_k = 0$ for $k \in \mathcal{K} \setminus \bar{\mathcal{S}}$. Let $\epsilon^k = \bar{\epsilon}^k$ for $k \in \bar{\mathcal{S}}$ and $\epsilon^k = \hat{\mathbf{x}}^k - \mathbf{x}$ for $k \in \mathcal{K} \setminus \bar{\mathcal{S}}$, where $\mathbf{x} \in \mathcal{X}$ satisfying $\mathbf{a}^{i'}\mathbf{x} = b_i$, $\forall i \in \bar{\mathcal{I}}$. We want to show that $(\bar{\mathbf{v}}, \bar{\mathbf{u}}, \{\epsilon^k\}_{k \in \mathcal{K}})$ is feasible for Model (6). From (5b) and (5e), we have $\sum_{i \in \bar{\mathcal{I}}} \bar{y}_i \mathbf{a}^{i'}(\hat{\mathbf{x}}^k - \epsilon^k) = \sum_{i \in \bar{\mathcal{I}}} b_i \bar{y}_i$ or, equivalently, $\sum_{i \in \bar{\mathcal{I}}} \bar{y}_i(\mathbf{a}^{i'}(\hat{\mathbf{x}}^k - \epsilon^k) - b_i) = 0$ for each $k \in \bar{\mathcal{S}}$. Because $\bar{y}_i > 0$ for all $i \in \bar{\mathcal{I}}$, for each $k \in \bar{\mathcal{S}}$, we have $\mathbf{a}^{i'}(\hat{\mathbf{x}}^k - \epsilon^k) = b_i$ for $i \in \bar{\mathcal{I}}$ (i.e., $i: \bar{v}_i = 1$) and $\mathbf{a}^{i'}(\hat{\mathbf{x}}^k - \epsilon^k) \geq b_i$ for $i \in \mathcal{I} \setminus \bar{\mathcal{I}}$ (i.e., $i: \bar{v}_i = 0$). Also, for each $k \in \mathcal{K} \setminus \bar{\mathcal{S}}$, based on how ϵ^k is defined earlier, we also have $\mathbf{a}^{i'}(\hat{\mathbf{x}}^k - \epsilon^k) = b_i$ for all $i \in \bar{\mathcal{I}}$ and $\mathbf{a}^{i'}(\hat{\mathbf{x}}^k - \epsilon^k) \geq b_i$ for all $i \in \mathcal{I}$. Therefore, $\{\epsilon^k\}_{k \in \mathcal{K}}$ and $\bar{\mathbf{v}}$ together satisfy (6b). Because $\bar{\mathbf{u}}$ is constructed based on whether $\|\bar{\epsilon}^k\|_\ell \leq \tau$, $\{\epsilon^k\}_{k \in \mathcal{K}}$ and $\bar{\mathbf{u}}$ clearly satisfy (6c). Also, $\sum_{k=1}^K \bar{u}_k = \sum_{k \in \bar{\mathcal{S}}} \bar{u}_k = |\bar{\mathcal{S}}| \geq \theta K$, satisfying (6d). Thus, $(\bar{\mathbf{v}}, \bar{\mathbf{u}}, \{\epsilon^k\}_{k \in \mathcal{K}})$ is feasible for Model (6). Finally, from (5b), $\bar{\mathbf{c}} = \mathbf{A}'\bar{\mathbf{y}} = \sum_{i \in \bar{\mathcal{I}}} \bar{y}_i \mathbf{a}^i = \sum_{i: \bar{v}_i = 1} \bar{y}_i \mathbf{a}^i$, and therefore, $\bar{\mathbf{c}} \in \text{cone}(\{\mathbf{a}^i\}_{i: \bar{v}_i = 1})$. Because $\bar{\mathbf{c}}$ is feasible for $\mathbf{QIO}(\mathcal{K}, \tau, \theta)$, $\|\bar{\mathbf{c}}\|_p = 1$.

($\Leftarrow$) Assume $(\bar{\mathbf{v}} \neq \mathbf{0}, \bar{\mathbf{u}}, \{\epsilon^k\}_{k \in \mathcal{K}})$ is feasible for Model (6). Let $\bar{\mathcal{I}} = \{i \in \mathcal{I} \mid \bar{v}_i = 1\}$ and $\bar{\mathcal{S}} = \{k \in \mathcal{K} \mid \bar{u}_k = 1\}$ and consider $\bar{\mathbf{c}} \in \text{cone}(\{\mathbf{a}^i\}_{i \in \bar{\mathcal{I}}})$ such that $\|\bar{\mathbf{c}}\|_p = 1$. We can write $\bar{\mathbf{c}} = \sum_{i \in \bar{\mathcal{I}}} \bar{\lambda}_i \mathbf{a}^i$, where $\bar{\lambda}_i \geq 0$ $\forall i \in \bar{\mathcal{I}}$. If we let $\bar{y}_i = \bar{\lambda}_i$ for all $i \in \bar{\mathcal{I}}$ and $\bar{y}_i = 0$ for all $i \in \mathcal{I} \setminus \bar{\mathcal{I}}$, then $\bar{\mathbf{c}} = \sum_{i \in \bar{\mathcal{I}}} \bar{y}_i \mathbf{a}^i = \mathbf{A}'\bar{\mathbf{y}}$ satisfies (5b); because $\bar{\mathbf{y}} \geq \mathbf{0}$, (5c) is also satisfied. Because $\mathbf{a}^{i'}(\hat{\mathbf{x}}^k - \bar{\epsilon}^k) \geq b_i$ holds for all $i \in \mathcal{I}, k \in \mathcal{K}$ by (6b), (5d) is satisfied for all $k \in \mathcal{K}$. Also because Model (6) is feasible, $\mathbf{a}^{i'}(\hat{\mathbf{x}}^k - \bar{\epsilon}^k) = b_i$ for all $i \in \bar{\mathcal{I}}$ and all k for which $\bar{u}_k = 1$, that is, $k \in \bar{\mathcal{S}}$; multiplying both sides by $\bar{y}_i$ and summing over all $i \in \mathcal{I}$, we have $\sum_{i \in \mathcal{I}} \bar{y}_i \mathbf{a}^{i'}(\hat{\mathbf{x}}^k - \bar{\epsilon}^k) = \sum_{i \in \mathcal{I}} \bar{y}_i b_i$ or equivalently, $\bar{\mathbf{c}}'(\hat{\mathbf{x}}^k - \bar{\epsilon}^k) = \mathbf{b}'\bar{\mathbf{y}}$, satisfying (5e). Clearly, $\{\bar{\epsilon}_k\}_{k \in \mathcal{K}}$ and $\bar{\mathcal{S}}$ satisfy Constraints (5f)–(5h). Therefore, $(\bar{\mathbf{c}}, \{\bar{\epsilon}^k\}_{k \in \mathcal{K}}, \bar{\mathbf{y}}, \bar{\mathcal{S}})$ is feasible for $\mathbf{QIO}(\mathcal{K}, \tau, \theta)$. □

Proof of Proposition 3. For this proof, we first refer the reader to Appendix A.3 in which we show a formulation to compute the true inverse stability measure; this formulation is nonconvex, but we use it for this proof. Note that the true inverse stability measure ξ^* is the optimal value of Problem (A.3) in Appendix A.3.

We first present a variant of (A.3) for each $i \in \bar{\mathcal{I}}$ separately (i.e., $|\bar{\mathcal{I}}|$ of them) and show that the optimal objective value of (A.3), that is, ξ^*, is bounded below by the largest optimal objective value among these $|\bar{\mathcal{I}}|$ smaller problems. Consider the following problem for each $i \in \bar{\mathcal{I}}$:

$$\underset{\{\mathbf{x}^k\}_{k \in \mathcal{K}}, \mathcal{S}^i}{\text{minimize}} \quad \sum_{k \in \mathcal{K}} \|\mathbf{x}^k - \hat{\mathbf{x}}^k\|_\ell \tag{C.1a}$$

$$\text{subject to} \quad D^i(\mathbf{x}^k) = \min_{\mathbf{z} \in \mathcal{X}_i} \{\|\mathbf{x}^k - \mathbf{z}\|_\ell\}, \quad \forall k \in \mathcal{K}, \tag{C.1b}$$

$$D^i(\mathbf{x}^k) \geq \tau + \sigma, \quad \forall k \in \mathcal{S}^i, \tag{C.1c}$$

$$|\mathcal{S}^i| \geq \lfloor (1-\theta)K \rfloor + 1. \tag{C.1d}$$

Let ζ_i^* be the optimal value of this problem for each $i \in \bar{\mathcal{I}}$. Note that, unlike in Formulation (A.3), in which constraints of $D^i(\mathbf{x}^k) \geq \tau + \sigma$ in (A.3c) must be satisfied for all $i \in \hat{\mathcal{I}}$, Problem (C.1) requires Constraint (C.1c) to be satisfied for only one $i \in \bar{\mathcal{I}} \subseteq \hat{\mathcal{I}}$. Hence, we have $\zeta_i^* \leq \xi^*$ for all $i \in \bar{\mathcal{I}}$ or, equivalently, $\max_{i \in \bar{\mathcal{I}}} \{\zeta_i^*\} \leq \xi^*$. Problem (C.1) is still nonconvex, and thus, finding ζ_i^* is still a challenge. We next show that its optimal objective value ζ_i^* is equal to the sum of the optimal objective values of even smaller problems that can be efficiently solved. Given $i \in \bar{\mathcal{I}}$ and $k \in \mathcal{K}$, consider the following problem:

$$\underset{\mathbf{x}^k}{\text{minimize}} \quad \|\mathbf{x}^k - \hat{\mathbf{x}}^k\|_\ell \tag{C.2a}$$

$$\text{subject to} \quad D^i(\mathbf{x}^k) = \min_{\mathbf{z} \in \mathcal{X}_i} \{\|\mathbf{x}^k - \mathbf{z}\|_\ell\}, \tag{C.2b}$$

$$D^i(\mathbf{x}^k) \geq \tau + \sigma. \tag{C.2c}$$

Let $\tilde{\zeta}_{ik}$ denote the optimal value of this problem. For a given i, let $\tilde{\zeta}_{i[k]}$ be the kth smallest value of $\tilde{\zeta}_{ik}, k \in \mathcal{K}$. Then, we have $\zeta_i^* = \sum_{k=1}^{\lfloor (1-\theta)K \rfloor + 1} \tilde{\zeta}_{i[k]}$. Thus, $\max_{i \in \bar{\mathcal{I}}} \{\sum_{k=1}^{\lfloor (1-\theta)K \rfloor + 1} \tilde{\zeta}_{i[k]}\} = \max_{i \in \bar{\mathcal{I}}} \{\zeta_i^*\} \leq \xi^*$.

Next, we show how we can find the optimal value for Formulation (C.2), that is, $\tilde{\zeta}_{ik}$. Let $d_{ik}^* = \min_{\mathbf{x} \in \mathcal{X}_i} \{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\}$ and $\bar{\tau} = \tau + \sigma$. Note that, when $d_{ik}^* \geq \bar{\tau}$, we have $\tilde{\zeta}_{ik} = 0$ because $\mathbf{x}^k = \hat{\mathbf{x}}^k$ satisfies both (C.2b) and (C.2c). Now, we want to show that, if $d_{ik}^* < \bar{\tau}$, $\tilde{\zeta}_{ik} = \bar{\tau} - d_{ik}^*$. To do so, we first show that $\tilde{\zeta}_{ik} \geq \bar{\tau} - d_{ik}^*$. Let $\bar{\mathbf{x}} \in \text{argmin}_{\mathbf{x} \in \mathcal{X}_i}\{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\}$. Supposed to the contrary that there exists an optimal solution $\tilde{\mathbf{x}}^k$ that attains $\tilde{\zeta}_{ik} < \bar{\tau} - d_{ik}^*$. By the definition of d_{ik}^*, we can rewrite this inequality as $\|\tilde{\mathbf{x}}^k - \hat{\mathbf{x}}^k\|_\ell < \bar{\tau} - \|\hat{\mathbf{x}}^k - \bar{\mathbf{x}}\|_\ell$ or, equivalently, $\|\tilde{\mathbf{x}}^k - \hat{\mathbf{x}}^k\|_\ell + \|\hat{\mathbf{x}}^k - \bar{\mathbf{x}}\|_\ell < \bar{\tau}$. Furthermore, by triangular inequality the following is true: $\|\tilde{\mathbf{x}}^k - \bar{\mathbf{x}}\|_\ell \leq \|\tilde{\mathbf{x}}^k - \hat{\mathbf{x}}^k\|_\ell + \|\hat{\mathbf{x}}^k - \bar{\mathbf{x}}\|_\ell < \bar{\tau}$. However, because $\bar{\mathbf{x}} \in \mathcal{X}_i$, we have $D^i(\tilde{\mathbf{x}}^k) = \min_{\mathbf{z} \in \mathcal{X}_i}\{\|\tilde{\mathbf{x}}^k - \mathbf{z}\|_\ell\} \leq \|\tilde{\mathbf{x}}^k - \bar{\mathbf{x}}\|_\ell < \bar{\tau}$, which violates (C.2c) and, thus, contradicts the feasibility of $\tilde{\mathbf{x}}^k$. Thus, it must be that $\tilde{\zeta}_{ik} \geq \bar{\tau} - d_{ik}^*$. All we need now is to show that there exists a feasible solution $\tilde{\mathbf{x}}^k$ to Problem (C.2) that attains the objective function value of $\bar{\tau} - d_{ik}^*$. Note that, if a given $\tilde{\mathbf{x}}^k$ is feasible for Problem (C.2), we must have $D^i(\tilde{\mathbf{x}}^k) = \min_{\mathbf{z} \in \mathcal{X}_i}\{\|\tilde{\mathbf{x}}^k - \mathbf{z}\|_\ell\}$ bounded below by $\bar{\tau}$ (because of (C.2c)). Consider $\tilde{\mathbf{x}}^k = \frac{\bar{\tau}}{d_{ik}^*}\hat{\mathbf{x}}^k - \frac{\bar{\tau}}{d_{ik}^*}\bar{\mathbf{x}} + \bar{\mathbf{x}}$. Clearly,

$\mathbf{x}^k = \tilde{\mathbf{x}}^k$ and $\mathbf{z} = \bar{\mathbf{x}}$ together satisfy all constraints in (C.2) with equality because we have $\bar{\mathbf{x}} \in \mathcal{X}_i$ and

$$\|\tilde{\mathbf{x}}^k - \bar{\mathbf{x}}\|_\ell = \|\frac{\bar{\tau}}{d^*_{ik}}\hat{\mathbf{x}}^k - \frac{\bar{\tau}}{d^*_{ik}}\bar{\mathbf{x}} + \bar{\mathbf{x}} - \bar{\mathbf{x}}\|_\ell \tag{C.3a}$$

$$= \frac{\bar{\tau}}{d^*_{ik}}\|\hat{\mathbf{x}}^k - \bar{\mathbf{x}}\|_\ell \tag{C.3b}$$

$$= \frac{\bar{\tau}}{d^*_{ik}}d^*_{ik} = \bar{\tau}. \tag{C.3c}$$

Moreover, $\mathbf{x}^k = \tilde{\mathbf{x}}^k$ leads to the objective value $\tilde{\zeta}_{ik} = \|\tilde{\mathbf{x}}^k - \hat{\mathbf{x}}^k\|_\ell = \|\frac{\bar{\tau}}{d^*_{ik}}\hat{\mathbf{x}}^k - \frac{\bar{\tau}}{d^*_{ik}}\bar{\mathbf{x}} + \bar{\mathbf{x}} - \hat{\mathbf{x}}^k\|_\ell = \frac{\bar{\tau}-d^*_{ik}}{d^*_{ik}}\|\hat{\mathbf{x}}^k - \bar{\mathbf{x}}\|_\ell = \bar{\tau} - d^*_{ik}$ and, thus, is an optimal solution to (C.2).

As a result, $\max_{i\in\bar{\mathcal{I}}}\{\sum_{k=1}^{(\lfloor(1-\theta)K\rfloor+1)}\max(0,\bar{\tau}-d^*_{i[k]})\} = \max_{i\in\bar{\mathcal{I}}}\{\sum_{k=1}^{(\lfloor(1-\theta)K\rfloor+1)}\tilde{\zeta}_{i[k]}\} = \max_{i\in\bar{\mathcal{I}}}\{\tilde{\zeta}^*_i\} \leq \xi^*$. Clearly, because $\tau < \bar{\tau}$, we have $\max_{i\in\bar{\mathcal{I}}}\{\sum_{k=1}^{(\lfloor(1-\theta)K\rfloor+1)}\max(0,\tau-d^*_{i[k]})\} \leq \xi^*$ as desired. □

Proof of Proposition 4. Given a feasible solution $(\bar{\mathbf{v}},\bar{\mathbf{u}},\{\bar{\epsilon}^k\}_{k\in\mathcal{K}})$ for (6), let $\bar{\mathcal{I}} = \{i\in\mathcal{I} \mid \bar{v}_i = 1\}$ and $\bar{\mathcal{S}} = \{k\in\mathcal{K} \mid \bar{u}_k = 1\}$. Because of Constraints (6b) and (6c), for each $k\in\bar{\mathcal{S}}$, we have $\hat{\mathbf{x}}^k - \bar{\epsilon}^k \in \{\mathbf{x}\in\mathcal{X} \mid \mathbf{a}^{i'}\mathbf{x}^k = b_i, \forall i\in\bar{\mathcal{I}}\} = \cap_{i\in\bar{\mathcal{I}}}\mathcal{X}_i$ and $\|\bar{\epsilon}^k\|_\ell \leq \tau$. Now, consider $\bar{\mathbf{c}} \in \text{cone}_+(\{\mathbf{a}^i\}_{i:\bar{v}_i=1})$. This $\bar{\mathbf{c}}$ vector leads to $\mathcal{X}^*(\bar{\mathbf{c}}) = \arg\min \mathbf{FO}(\bar{\mathbf{c}}) = \cap_{i\in\bar{\mathcal{I}}}\mathcal{X}_i$. Recall that $\hat{\mathbf{x}}^k - \bar{\epsilon}^k \in \cap_{i\in\bar{\mathcal{I}}}\mathcal{X}_i$; this means that $\hat{\mathbf{x}}^k - \bar{\epsilon}^k \in \mathcal{X}^*(\bar{\mathbf{c}})$ for all $k\in\bar{\mathcal{S}}$. Moreover, because $\mathcal{X}$ is an n-dimensional polytope and $|\bar{\mathcal{I}}| = n$, $\mathcal{X}^*(\bar{\mathbf{c}})$ is a singleton, that is, $\mathcal{X}^*(\bar{\mathbf{c}}) = \{\hat{\mathbf{x}}^k - \bar{\epsilon}^k\}$, $\forall k\in\bar{\mathcal{S}}$ and $\hat{\mathbf{x}}^k - \bar{\epsilon}^k = \hat{\mathbf{x}}^{k'} - \bar{\epsilon}^{k'}$ for all $k,k'\in\mathcal{K}$. To complete the proof, for all $k\in\bar{\mathcal{S}}$, we have $\max_{\mathbf{x}\in\mathcal{X}^*(\bar{\mathbf{c}})}\{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\} = \|\hat{\mathbf{x}}^k - (\hat{\mathbf{x}}^k - \bar{\epsilon}^k)\|_\ell = \|\bar{\epsilon}^k\|_\ell \leq \tau$. □

Proof of Proposition 5. Let $\mathcal{H}_1 = \{\mathbf{a}^i : v_i^1 = 1\}$ and $\mathcal{H}_2 = \{\mathbf{a}^i : v_i^2 = 1\}$. Because $\mathbf{v}^1 \geq \mathbf{v}^2$, we have $\mathcal{H}_2 \subseteq \mathcal{H}_1$, and thus, $\mathcal{C}_2 \subseteq \mathcal{C}_1$. Given the data set $\hat{\mathcal{X}}$, we want to show that, for any $\mathbf{c}^2\in\mathcal{C}_2$, there exists $\mathbf{c}^1\in\mathcal{C}_1$ such that $\max_{\mathbf{x}\in\mathcal{X}^*(\mathbf{c}^1)}\{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\} \leq \max_{\mathbf{x}\in\mathcal{X}^*(\mathbf{c}^2)}\{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\}$, $\forall k : \bar{u}_k = 1$. Suppose to the contrary that there exists $\mathbf{c}^*\in\mathcal{C}_2$ such that $\max_{\mathbf{x}\in\mathcal{X}^*(\mathbf{c}^*)}\{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\} < \max_{\mathbf{x}\in\mathcal{X}^*(\mathbf{c})}\{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\}$, $\forall \mathbf{c}\in\mathcal{C}_1$ for some $k : \bar{u}_k = 1$. This means $\mathbf{c}^* \in \mathcal{C}_2\backslash\mathcal{C}_1$, and thus, $\mathcal{C}_2\backslash\mathcal{C}_1 \neq \emptyset$, which contradicts $\mathcal{C}_2 \subseteq \mathcal{C}_1$. Thus, for any $\mathbf{c}^2\in\mathcal{C}_2$, there exists $\mathbf{c}^1\in\mathcal{C}_1$ such that $\max_{\mathbf{x}\in\mathcal{X}^*(\mathbf{c}^1)}\{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\} \leq \max_{\mathbf{x}\in\mathcal{X}^*(\mathbf{c}^2)}\{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\}$, $\forall k : \bar{u}_k = 1$ as desired. □

Proof of Proposition 6. Let $\mathcal{H}_1 = \{\mathbf{a}^i : v_i^1 = 1\}$ and $\mathcal{H}_2 = \{\mathbf{a}^i : v_i^2 = 1\}$. Because $\mathbf{v}^1 \geq \mathbf{v}^2$, we have $\mathcal{H}_2 \subseteq \mathcal{H}_1$ and $\mathcal{C}_2 \subseteq \mathcal{C}_1$. Because $\mathcal{C}_2 \subseteq \mathcal{C}_1$,

$$\min_{\tilde{\mathcal{X}}}\{d(\hat{\mathcal{X}},\tilde{\mathcal{X}}) \mid \mathcal{C}_1 \cap \mathcal{C}(\tilde{\mathcal{X}}) = \emptyset\}$$

$$= \min_{\tilde{\mathcal{X}}}\{d(\hat{\mathcal{X}},\tilde{\mathcal{X}}) \mid (\mathcal{C}_2 \cap \mathcal{C}(\tilde{\mathcal{X}})) \cup ((\mathcal{C}_1\backslash\mathcal{C}_2) \cap \mathcal{C}(\tilde{\mathcal{X}})) = \emptyset\}$$

$$= \min_{\tilde{\mathcal{X}}}\{d(\hat{\mathcal{X}},\tilde{\mathcal{X}}) \mid \mathcal{C}_2 \cap \mathcal{C}(\tilde{\mathcal{X}}) = \emptyset \text{ and } (\mathcal{C}_1\backslash\mathcal{C}_2) \cap \mathcal{C}(\tilde{\mathcal{X}}) = \emptyset\}$$

$$\geq \min_{\tilde{\mathcal{X}}}\{d(\hat{\mathcal{X}},\tilde{\mathcal{X}}) \mid \mathcal{C}_2 \cap \mathcal{C}(\tilde{\mathcal{X}}) = \emptyset\}, \text{ as desired. } \square$$

Proof of Lemma 2. Assume that there exists an optimal solution $(\bar{\mathbf{v}},\bar{\mathbf{u}},\{\bar{\epsilon}^k\}_{k\in\mathcal{K}})$ for **MQIO**$(\mathcal{K},\tau,\theta)$ such that $\sum_{k=1}^K \bar{u}_k > \lceil\theta K\rceil$. Without loss of generality let $\sum_{k=1}^K \bar{u}_k = \lceil\theta K\rceil + r$, where $r > 0$. Pick any r observations for which $\bar{u}_k = 1$ and change their values to zero, and denote this new vector by $\tilde{\mathbf{u}}$. Then, we can construct a new solution $(\bar{\mathbf{v}},\tilde{\mathbf{u}},\{\bar{\epsilon}^k\}_{k\in\mathcal{K}})$, which is still feasible for **MQIO**$(\mathcal{K},\tau,\theta)$ because $\|\bar{\epsilon}^k\|_\ell \leq \tau + M_2(1-\bar{u}_k) \leq \tau + M_2(1-\tilde{u}_k)$, $\forall k\in\mathcal{K}$ and $\sum_{k=1}^K \tilde{u}_k = \lceil\theta K\rceil \geq \theta K$, and generates the same objective value; hence, it is also an optimal solution. □

Proof of Proposition 8. Consider a matrix $\bar{\mathbf{D}}$ constructed as in (9). By definition, each row $k\in\mathcal{K}$ of $\bar{\mathbf{D}}$ is a binary vector $\mathbf{v}^{k'}$, and there exists $\bar{\epsilon}^k$ such that $(\mathbf{v}^k,\bar{\epsilon}^k)$ is feasible for (10). Assume that there exists an all-one submatrix in $\bar{\mathbf{D}}$ whose rows and columns correspond to $\bar{\mathcal{S}}$ and $\bar{\mathcal{A}}$, respectively, such that $\bar{\mathcal{S}} \subseteq \mathcal{K}$, $|\bar{\mathcal{S}}| \geq \theta K$, and $\bar{\mathcal{A}} \subseteq \mathcal{I}$. Define $\bar{\mathbf{v}}$ such that $\bar{v}_i = 1$ for all $i\in\bar{\mathcal{A}}$ and $\bar{v}_i = 0$ otherwise. Then, we have $\bar{\mathbf{v}} \leq \mathbf{v}^k$, $\forall k\in\bar{\mathcal{S}}$, and thus, $(\bar{\mathbf{v}},\bar{\epsilon}^k)$ is also feasible for (10) for all $k\in\bar{\mathcal{S}}$. Define $\bar{\mathbf{u}}$ such that $\bar{u}_k = 1$ for all $k\in\bar{\mathcal{S}}$ and $\bar{u}_k = 0$ otherwise; note that $\sum_{k=1}^K \bar{u}_k \geq \theta K$. Let $\epsilon^k = \bar{\epsilon}^k$ for $k\in\bar{\mathcal{S}}$ and $\epsilon^k = \hat{\mathbf{x}}^k - \mathbf{x}^k$ for $k\in\mathcal{S}\backslash\bar{\mathcal{S}}$, where $\mathbf{x}^k$ is arbitrarily chosen from $\{\mathbf{x}\in\mathcal{X} \mid \mathbf{a}^i\mathbf{x} = b_i, \forall i\in\bar{\mathcal{A}}\}$. We first show that the solution $(\bar{\mathbf{v}},\bar{\mathbf{u}},\{\epsilon^k\}_{k\in\mathcal{K}})$ constructed as before is feasible for Problem (6). For all $i\in\bar{\mathcal{A}} = \{i \mid \bar{v}_i = 1\}$, we have $\mathbf{a}^{i'}(\hat{\mathbf{x}}^k - \epsilon^k) = \mathbf{a}^{i'}\mathbf{x}^k = b_i$ for each $k\in\mathcal{K}$. Also, for all $i\in\mathcal{I}\backslash\bar{\mathcal{A}}$ (i.e., $\{i \mid \bar{v}_i = 0\}$), we have $\mathbf{a}^{i'}(\hat{\mathbf{x}}^k - \epsilon^k) = \mathbf{a}^{i'}\mathbf{x}^k \geq b_i$ for each $k\in\mathcal{K}$. Thus, $\bar{\mathbf{v}}$ and $\{\epsilon^k\}_{k\in\mathcal{K}}$ satisfy (6b). Constraint (6c) is satisfied because $\|\epsilon^k\|_\ell = \|\bar{\epsilon}^k\|_\ell \leq \tau$ for all $k\in\bar{\mathcal{S}} = \{k \mid \bar{u}_k = 1\}$. Constraint (6d) is satisfied as we already know $\sum_{k=1}^K \bar{u}_k \geq \theta K$. As a result, $(\bar{\mathbf{v}},\bar{\mathbf{u}},\{\epsilon^k\}_{k\in\mathcal{K}})$ is feasible for (6). Then, by Theorem 1, there must also exist a solution $(\bar{\mathbf{c}},\{\epsilon^k\}_{k\in\mathcal{K}},\bar{\mathbf{y}},\bar{\mathcal{S}})$ feasible for **QIO**$(\mathcal{K},\tau,\theta)$, where $\bar{\mathbf{c}} \in \text{cone}(\{\mathbf{a}^i\}_{i:\bar{v}_i=1}) = \text{cone}(\{\mathbf{a}^i\}_{i\in\bar{\mathcal{A}}})$ as desired. □

Proof of Proposition 9. Let $P(r,k)$ denote the optimization problem solved in line 5 of Algorithm 2 with respect to $\hat{\mathbf{x}}^k$, and $(\mathbf{v}^{(r,k)},\epsilon^{(r,k)}) \in \arg\max_{(\mathbf{v},\epsilon)}P(r,k)$. We first show that there exists $\bar{r} \leq \sum_{z=z^*}^n |\mathcal{V}_z|$ such that $\mathbf{v}^{(\bar{r},1)} = \mathbf{v}^*$ for problem $P(\bar{r},1)$ (i.e., with respect to the first data point $\hat{\mathbf{x}}^1$). For each iteration $r \geq 2$, each cut only prevents the solution found in the previous iteration $r - 1$, that is, $\mathbf{v}^{(r-1,1)}$, from being feasible for Constraints (10b)–(10d). This means that $P(r,1)$ can be solved $\sum_{z=1}^n |\mathcal{V}_z|$ times with iteratively added cuts to collect all solutions satisfying (10b)–(10d) with respect to $\hat{\mathbf{x}}^1$. Note that because $(\mathbf{v}^*,\epsilon^{k*})$ satisfies (6b) for all $k\in\mathcal{K}$ and $u_1^* = 1$, $(\mathbf{v}^*,\epsilon^{1*})$ satisfies Constraints (10b)–(10d) with respect to $\hat{\mathbf{x}}^1$. Thus, it is guaranteed that, for some $\bar{r} \leq \sum_{z=1}^n |\mathcal{V}_z|$, we have $\mathbf{v}^{(\bar{r},1)} = \mathbf{v}^*$. Furthermore, because the value function of $P(r,1)$ is nonincreasing in r, for any $r > \sum_{z=z^*}^n |\mathcal{V}_z|$, the optimal value of $P(r,1)$ is less than z^*, and thus, $\mathbf{v}^*$ is not achievable. Therefore, $\bar{r}$ must be no greater than $\sum_{z=z^*}^n |\mathcal{V}_z|$.

Next, we need to show that, when $\mathbf{v}^{(\bar{r},1)} = \mathbf{v}^*$ at iteration $\bar{r}$, the algorithm achieves $\mathbf{v}^{Alg} = \mathbf{v}^*$ and $z^{Alg} = z^*$. To do so, we first show that, when $\mathbf{v}^{(\bar{r},1)} = \mathbf{v}^*$, lines 4–9 of the

algorithm lead to $\bar{\mathbf{D}}_{\bar{r}}$ with an all-one submatrix with z^* columns and at least $\lceil\theta K\rceil$ rows. Let $\mathcal{I}^* = \{i \in \mathcal{I} \mid v_i^* = 1\}$ and $\mathcal{S}^* = \{k \in \mathcal{K} \mid u_k^* = 1\}$. For some given r and k, $P(r, k)$ has an optimal solution with $v_i^{(r,k)} = 0$ if $w_i^k = 0$; also, based on how the weight parameters $\mathbf{w}^k$ are updated (see Section 4.3), if $v_i^{(r,k)} = 0$ for $k = 1,\ldots,k'-1$, we have $w_i^{k'} = 0$. Generalizing this observation, if $\mathbf{v}^{(\bar{r},1)} = \mathbf{v}^*$, then for all $k = 2,\ldots,K$, we have $w_i^k = 0$ for all $i \in \mathcal{I} \setminus \mathcal{I}^*$ and $w_i^k > 0$ for all $i \in \mathcal{I}^*$, and thus, the maximum of the objective function of $P(\bar{r},k)$ for each k, that is, $\mathbf{w}^{k'}\mathbf{v}$, can be obtained when $\mathbf{v} = \mathbf{v}^*$. Because $(\mathbf{v}^*, \boldsymbol{\epsilon}^{k*})$ satisfies all constraints of Problem (10) as well as all cuts in $P(\bar{r},k)$ generated up to iteration $\bar{r}$ for all $k \in \mathcal{S}^*$, we have $(\mathbf{v}^*, \boldsymbol{\epsilon}^{k*}) \in \mathrm{argmax}_{\mathbf{v},\epsilon} P(\bar{r},k)$. This means, if $\mathbf{v}^{*'}$ is assigned to the first row of $\bar{\mathbf{D}}_{\bar{r}}$, it is guaranteed that $\mathbf{v}^{*'}$ is also assigned to rows $k \in \mathcal{S}^*$ of $\bar{\mathbf{D}}_{\bar{r}}$. Clearly, the submatrix of this $\bar{\mathbf{D}}_{\bar{r}}$ with columns and rows corresponding to $i \in \mathcal{I}^*$ and $k \in \mathcal{S}^*$, respectively, is all-one, and we have $|\mathcal{I}^*| = z^*$ and $|\mathcal{S}^*| \geq \theta K$. The clique problem in line 10 of the algorithm (also see Formulation (B.1)) then takes this $\bar{\mathbf{D}}_{\bar{r}}$ as input to return its optimal solution $\bar{\mathbf{v}}^{\bar{r}} = \mathbf{v}^*$ with $\bar{\mathbf{v}}^{r'}\mathbf{e} = z^*$. Because z^* is the maximum achievable value (it is the optimal value of the exact MQIO problem), it satisfies the condition in line 12 of Algorithm 2, and thus, we have $\mathbf{v}^{Alg} = \bar{\mathbf{v}}^{\bar{r}} = \mathbf{v}^*$ and $z^{Alg} = z^*$.

Finally, we show that the stopping criterion is valid; that is, the while loop condition is not violated before $\mathbf{v}^{Alg} = \mathbf{v}^*$ is observed. Consider a certain iteration $\bar{r} \leq \sum_{z=z^*}^n |\mathcal{V}_z|$ and suppose we have $\mathbf{v}^{Alg} \neq \mathbf{v}^*$ and $z^{Alg} < z^*$ at the end of iteration $\bar{r}$ (i.e., lines 13 and 14 of Algorithm 2). For the while loop condition to be violated in the next iteration, that is, iteration $\bar{r}+1$, we must have $|\mathcal{G}_{\bar{r}+1}| < z^{Alg} < z^*$, which means $\mathbf{e}'\mathbf{v}^{(\bar{r},1)} < z^*$. However, this cannot happen because we show earlier in this proof that, for all $r \leq \sum_{z=z^*}^n |\mathcal{V}_z|$, we have $\mathbf{e}'\mathbf{v}^{(r,1)} \geq z^*$. $\square$

Proof of Proposition A.1. Proposition 2 implies that, given a solution $(\bar{\mathbf{c}}, \{\bar{\boldsymbol{\epsilon}}^k\}_{k\in\mathcal{K}}, \bar{\mathbf{y}}, \bar{\mathcal{S}})$ feasible for (A.1), $(\tilde{\mathbf{c}} = \mathbf{a}^{\tilde{i}}, \{\bar{\boldsymbol{\epsilon}}^k\}_{k\in\mathcal{K}}, \tilde{\mathbf{y}} = \mathbf{e}_{\tilde{i}}, \bar{\mathcal{S}})$ is also feasible for any $\tilde{i} \in \bar{\mathcal{I}} = \{i \in \mathcal{I} \mid \bar{y}_i > 0\}$. For an arbitrary $\tilde{i} \in \bar{\mathcal{I}}$, consider the following formulation defined for each $k \in \mathcal{K}$:

$$\underset{\boldsymbol{\epsilon}^k}{\mathrm{minimize}}\, \{\|\boldsymbol{\epsilon}^k\|_\ell \mid \mathbf{A}(\hat{\mathbf{x}}^k - \boldsymbol{\epsilon}^k) \geq \mathbf{b}, \mathbf{a}^{\tilde{i}'}(\hat{\mathbf{x}}^k - \boldsymbol{\epsilon}^k) = b_{\tilde{i}}\}. \quad (C.4)$$

Let $\tilde{\boldsymbol{\epsilon}}^k$ be the optimal solution to (C.4) for each $k \in \mathcal{K}$ (here, we suppress the index $\tilde{i}$ for brevity). Note that a solution constructed as $(\tilde{\mathbf{c}} = \mathbf{a}^{\tilde{i}}, \{\tilde{\boldsymbol{\epsilon}}^k\}_{k\in\mathcal{K}}, \tilde{\mathbf{y}} = \mathbf{e}_{\tilde{i}}, \bar{\mathcal{S}})$ is also feasible for (A.1) because $\|\tilde{\boldsymbol{\epsilon}}^k\|_\ell \leq \|\bar{\boldsymbol{\epsilon}}^k\|_\ell \leq \tau$ for each $k \in \bar{\mathcal{S}}$. The objective of (A.1), that is, minimizing τ, is equivalent to minimizing $\max_{k\in\bar{\mathcal{S}}}\{\|\boldsymbol{\epsilon}^k\|_\ell\}$. Because $\|\tilde{\boldsymbol{\epsilon}}^k\|_\ell \leq \|\bar{\boldsymbol{\epsilon}}^k\|_\ell \leq \tau$ for each $k \in \bar{\mathcal{S}}$, we have $\max_{k\in\bar{\mathcal{S}}}\{\|\tilde{\boldsymbol{\epsilon}}^k\|_\ell\} \leq \max_{k\in\bar{\mathcal{S}}}\{\|\bar{\boldsymbol{\epsilon}}^k\|_\ell\}$. That is, for any given feasible solution $(\bar{\mathbf{c}}, \{\bar{\boldsymbol{\epsilon}}^k\}_{k\in\mathcal{K}}, \bar{\mathbf{y}}, \bar{\mathcal{S}})$ for (A.1) in which $\bar{\mathbf{c}}$ is not identical to $\mathbf{a}^i$ for any $i \in \mathcal{I}$, we can always construct another feasible solution $(\tilde{\mathbf{c}} = \mathbf{a}^i, \{\tilde{\boldsymbol{\epsilon}}^k\}_{k\in\mathcal{K}}, \tilde{\mathbf{y}} = \mathbf{e}_{\tilde{i}}, \bar{\mathcal{S}})$ without increasing the objective value of (A.1), which completes the proof. $\square$

Proof of Lemma A.1. Let $\hat{\mathcal{I}} = \{i \in \mathcal{I} \mid \mathbf{a}^i \in \hat{\mathcal{C}}\}$. Given shifted data $\tilde{\mathcal{X}}$, assume $\mathbf{c} = \mathbf{a}^i$ is not inverse-feasible for $\mathbf{QIO}(\tilde{\mathcal{X}}, \tau, \theta)$ for any $i \in \hat{\mathcal{I}}$. We want to show that there is no $\mathbf{c} \in \hat{\mathcal{C}}$ that is

inverse-feasible for $\mathbf{QIO}(\tilde{\mathcal{X}}, \tau, \theta)$. Suppose to the contrary that there is $\bar{\mathbf{c}} \in \hat{\mathcal{C}}$ feasible for $\mathbf{QIO}(\tilde{\mathcal{X}}, \tau, \theta)$ and let $\bar{\mathcal{I}}$ be such that $\bar{\mathbf{c}} \in \mathrm{cone}(\{\mathbf{a}^i\}_{i\in\bar{\mathcal{I}}})$. Then, from Proposition 2, there must be some $\bar{i} \in \bar{\mathcal{I}}$ such that $\mathbf{c} = \mathbf{a}^{\bar{i}}$ is also feasible for $\mathbf{QIO}(\tilde{\mathcal{X}}, \tau, \theta)$. Furthermore, because $\bar{\mathbf{c}} \in \hat{\mathcal{C}}$, this means that $\mathbf{c} = \mathbf{a}^{\bar{i}}$ must be also feasible for $\mathbf{QIO}(\hat{\mathcal{X}}, \tau, \theta)$ (i.e., it must be that $\bar{i} \in \hat{\mathcal{I}}$), which is a contradiction. $\square$

Proof of Proposition A.2. Because $\bar{\mathbf{c}}$ is a strict conic combination of the selected $\mathbf{a}^i$ vectors, we have $\mathcal{X}^*(\bar{\mathbf{c}}) = \cap_{i\in\bar{\mathcal{I}}}\mathcal{X}_i$. For each $k \in \bar{\mathcal{S}}$, let $\tilde{\mathbf{x}}^k \in \mathrm{argmin}_{\mathbf{x}\in\cap_{i\in\bar{\mathcal{I}}}\mathcal{X}_i}\{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\}$. Then, for all $k \in \bar{\mathcal{S}}$ and $i \in \bar{\mathcal{I}}$, we have $\max_{\mathbf{x}\in\mathcal{X}_i}\{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\} \leq \|\tilde{\mathbf{x}}^k - \hat{\mathbf{x}}^k\|_\ell + \max_{\mathbf{x}\in\mathcal{X}_i}\{\|\tilde{\mathbf{x}}^k - \mathbf{x}\|_\ell\} \leq \|\tilde{\mathbf{x}}^k - \hat{\mathbf{x}}^k\|_\ell + \rho_i$. Hence, for each $k \in \bar{\mathcal{S}}$,

$$
\begin{aligned}
\max_{\mathbf{x}\in\mathcal{X}^*(\bar{\mathbf{c}})}\{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\} &= \max_{\mathbf{x}\in\underset{i\in\bar{\mathcal{I}}}{\cap}\mathcal{X}_i}\{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\} \\
&\leq \min_{i\in\bar{\mathcal{I}}}\left\{\max_{\mathbf{x}\in\mathcal{X}_i}\{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\}\right\} \\
&\leq \min_{i\in\bar{\mathcal{I}}}\{\|\tilde{\mathbf{x}}^k - \hat{\mathbf{x}}^k\|_\ell + \rho_i\} \\
&= \min_{i\in\bar{\mathcal{I}}}\left\{\min_{\mathbf{x}\in\underset{i\in\bar{\mathcal{I}}}{\cap}\mathcal{X}_i}\{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\} + \rho_i\right\}, \\
&= \min_{\mathbf{x}\in\underset{i\in\bar{\mathcal{I}}}{\cap}\mathcal{X}_i}\{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\} + \min_{i\in\bar{\mathcal{I}}}\{\rho_i\}, \quad (C.5)
\end{aligned}
$$

where the first inequality holds because we have $\cap_{i\in\bar{\mathcal{I}}}\mathcal{X}_i \subseteq \mathcal{X}_i$ for all $i \in \bar{\mathcal{I}}$, which leads to $\max_{\mathbf{x}\in\cap_{i\in\bar{\mathcal{I}}}\mathcal{X}_i}\{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\} \leq \max_{\mathbf{x}\in\mathcal{X}_i}\{\|\hat{\mathbf{x}}^k - \mathbf{x}\|_\ell\}$, $\forall i \in \bar{\mathcal{I}}$. $\square$

Proof of Lemma B.1. Because $\bar{\alpha}_i^k = 0$ for all $i \in \bar{\mathcal{I}}$ and $(\bar{\boldsymbol{\alpha}}^k, \bar{\boldsymbol{\epsilon}}^k)$ is feasible for (B.2), we have $\mathbf{a}^{i'}(\hat{\mathbf{x}}^k - \bar{\boldsymbol{\epsilon}}^k) = b_i$ for all $i \in \bar{\mathcal{I}}$, $\mathbf{a}^{i'}(\hat{\mathbf{x}}^k - \bar{\boldsymbol{\epsilon}}^k) \geq b_i$ for all $i \in \mathcal{I} \setminus \bar{\mathcal{I}}$, and $\|\bar{\boldsymbol{\epsilon}}^k\|_\ell \leq \tau$. Let $\bar{v}_i^k = 1 - \bar{\alpha}_i^k$ for all $i \in \bar{\mathcal{I}}$ and $\bar{v}_i^k = 0$ otherwise. Clearly, $(\bar{\mathbf{v}}^k, \bar{\boldsymbol{\epsilon}}^k)$ satisfies all constraints of (10). $\square$

Proof of Proposition B.1. Assume that $\bar{\mathbf{D}}^{RP}$ satisfies (B.3) and has an all-one submatrix whose rows and columns correspond to $\bar{\mathcal{S}}$ and $\bar{\mathcal{A}}$, respectively, and $\bar{\mathcal{S}} \in \mathcal{K}$, $|\bar{\mathcal{S}}| \geq \theta K$, and $\bar{\mathcal{A}} \in \mathcal{I}$. By definition, for each $k \in \mathcal{K}$, there exists a feasible solution $(\bar{\boldsymbol{\alpha}}^k, \bar{\boldsymbol{\epsilon}}^k)$ for (B.2) with respect to $\hat{\mathbf{x}}^k$, where $\bar{\alpha}_i^k = 0$ for all $i \in \bar{\mathcal{A}}$ and $k \in \bar{\mathcal{S}}$, and $\bar{\alpha}_i^k > 0$ otherwise. This means that, by Lemma B.1, for each $k \in \mathcal{K}$, there exists a corresponding feasible solution $(\bar{\mathbf{v}}^k, \bar{\boldsymbol{\epsilon}}^k)$ for (10) with respect to $\hat{\mathbf{x}}^k$, where $\bar{v}_i^k = 1$ for all $i \in \bar{\mathcal{A}}$ and $k \in \bar{\mathcal{S}}$, and $\bar{v}_i^k = 0$ otherwise. Construct a matrix $\bar{\mathbf{D}} \in \{0,1\}^{K\times m}$ such that $\bar{\mathbf{D}}_{ki} = 1$ if $\bar{v}_i^k = 1$, and $\bar{\mathbf{D}}_{ki} = 0$ otherwise. Note that $\bar{\mathbf{D}}$ satisfies (9) and has an all-one submatrix whose rows and columns correspond to $\bar{\mathcal{S}}$ and $\bar{\mathcal{A}}$, respectively, and $\bar{\mathcal{S}} \subseteq \mathcal{K}$, $|\bar{\mathcal{S}}| \geq \theta K$, and $\bar{\mathcal{A}} \subseteq \mathcal{I}$. Thus, by Proposition 8, there exists a solution $(\bar{\mathbf{c}}, \{\bar{\boldsymbol{\epsilon}}^k\}_{k\in\mathcal{K}}, \bar{\mathbf{y}}, \bar{\mathcal{S}})$ feasible for $\mathbf{QIO}(\mathcal{K}, \tau, \theta)$, where $\bar{\mathbf{c}} \in \mathrm{cone}(\{\mathbf{a}^i\}_{i\in\bar{\mathcal{A}}})$. $\square$

Table D.1. Food Items and Nutrient Data per Serving

	Food type									Lower limit	Upper limit
	1	2	3	4	5	6	7	8	9		
Energy, KCAL	91.53	68.94	23.51	65.49	110.88	83.28	80.50	63.20	52.16	1,800.00	2,500.00
Total fat, g	4.95	0.71	1.80	3.48	6.84	4.41	5.80	0.94	0.18	44.00	78.00
Carbohydrate, g	6.89	12.16	0.25	0.00	5.44	4.68	0.56	11.42	13.59	220.00	330.00
Protein, g	4.90	3.68	1.59	7.99	6.80	5.93	6.27	2.40	0.41	56.00	NA
Fiber, g	0.00	0.06	0.00	0.00	0.28	0.29	0.00	1.19	1.81	20.00	30.00
Vitamin C, mg	0.01	1.76	0.00	0.00	0.17	0.16	0.00	0.02	11.19	90.00	2,000.00
Vitamin B6, mg	0.06	0.03	0.01	0.09	0.11	0.06	0.06	0.03	0.10	1.30	100.00
Vitamin B12, mcg	0.67	0.39	0.09	0.65	0.11	0.63	0.56	0.00	0.00	2.40	NA
Calcium, mg	172.09	125.72	46.24	2.21	5.90	15.03	29.00	27.21	6.14	1,000.00	2,500.00
Iron, mg	0.05	0.08	0.04	0.75	0.35	0.35	0.73	0.79	0.13	8.00	45.00
Copper, mg	0.02	0.04	0.01	0.03	0.03	0.03	0.04	0.05	0.06	0.90	10.00
Sodium, mg	61.02	48.24	65.08	72.32	211.05	128.27	223.50	125.62	1.42	1,500.00	2,300.00
Vitamin A, mcg	42.89	22.24	12.78	0.00	1.33	9.53	81.00	0.01	13.04	900.00	3,000.00
Max serving	4	8	4	5	5	5	4	4	8		

Appendix D. Supplemental Materials for the Numerical Results

D.1. The Diet Problem: Data
The data for the diet problem can be found in Table D.1.

D.2. The Transshipment Problem: Problem Description and Additional Numerical Results
Figure D.1 and the following formulation show the transshipment problem we consider.

$$\underset{\mathbf{x}^{(p)}, \mathbf{x}^{(t)}}{\text{minimize}} \quad \sum_{i \in \mathcal{N}_s} c_i^{(p)} x_i^{(p)} + \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} c_{ij}^{(t)} x_{ij}^{(t)} \tag{D.1a}$$

$$\text{subject to} \quad \sum_{j \in \mathcal{N} \backslash \mathcal{N}_s} x_{ij}^{(t)} = x_i^{(p)}, \quad \forall i \in \mathcal{N}_s, \tag{D.1b}$$

$$\sum_{i \in \mathcal{N} \backslash \mathcal{N}_d} x_{ij}^{(t)} = d_j, \quad \forall j \in \mathcal{N}_d, \tag{D.1c}$$

$$\sum_{j \in \mathcal{N}} x_{ij}^{(t)} - \sum_{j \in \mathcal{N}} x_{ji}^{(t)} = 0, \quad \forall i \in \mathcal{N}_t, \tag{D.1d}$$

$$0 \leq x_i^{(p)} \leq \pi_i^{(p)}, 0 \leq x_{ij}^{(t)} \leq \pi_{ij}^{(t)}, \quad \forall i, j \in \mathcal{N}, \tag{D.1e}$$

where $\mathcal{N}_s$, $\mathcal{N}_d$, and $\mathcal{N}_t$ denote the set of supply, demand, and transshipment nodes, respectively, and $\mathcal{N}$ denotes the set of all nodes. Variables $\mathbf{x}^{(p)}$ and $\mathbf{x}^{(t)}$ denote the production level at each supply node and transshipment flow on each arc, respectively. Parameters $\boldsymbol{\pi}^{(t)}$ and $\boldsymbol{\pi}^{(p)}$ represent the arc capacity and production capacity, respectively.

In addition to the numerical results presented in Section 5.4, we also compare the objective function values achieved by our forward solution $\bar{\mathbf{x}}^t$ (recall that $\mathbf{x} = [\mathbf{x}^{(p)}; \mathbf{x}^{(t)}]$ and $\mathbf{c} = [\mathbf{c}^{(p)}; \mathbf{c}^{(t)}]$) and those from the given data. We adopt the metrics from the previous study on online inverse LP (Bärmann et al. 2017); this

study assumes a single noiseless (i.e., optimal) decision at each time point and shows that both $(1/T)\sum_{t=1}^{T} \mathbf{c}^{true'}(\bar{\mathbf{x}}^t - \hat{\mathbf{x}}^t)$ and $(1/T)\sum_{t=1}^{T} \mathbf{c}^{t'}(\bar{\mathbf{x}}^t - \hat{\mathbf{x}}^t)$ converge to zero as $T \to \infty$. In our case, because we consider batches of data at each time, we use the following modified metrics: $(1/T)[\sum_{t=1}^{T} \sum_{k=1}^{K_t} \mathbf{c}^{true'}(\bar{\mathbf{x}}^t - \hat{\mathbf{x}}^k)/K_t]$ and $(1/T)[\sum_{t=1}^{T} \sum_{k=1}^{K_t} \mathbf{c}^{t'}(\bar{\mathbf{x}}^t - \hat{\mathbf{x}}^k)/K_t]$. Note that $\mathbf{c}^t$ is a cost vector randomly selected from the inverse set at time t. Figure D.2 shows these metrics achieved by our algorithm with noisy data sets at each iteration. Both metrics averaged over the data points also converge; however, neither of them reaches zero because of the noise in the data.

Figure D.1. Network for the Transshipment Example (Dong et al. 2018)
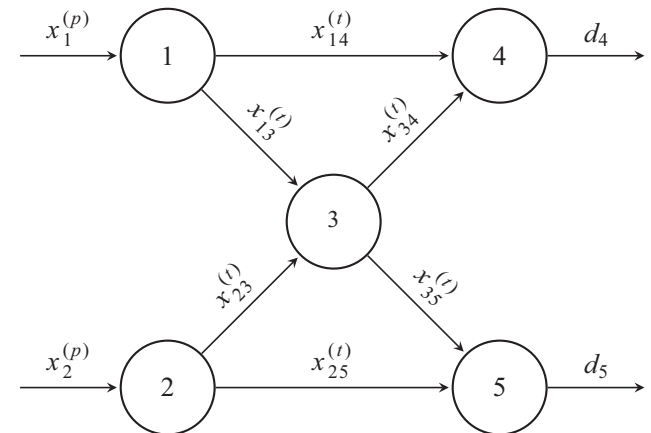
Notes. Nodes 1 and 2 are supply nodes and 4 and 5 are demand nodes. The capacity of each arc is 1.3, and production capacities are 3 and 1.5 for nodes 1 and 2, respectively. The true production costs are $c_1^{(p)} = 0.2393$ and $c_2^{(p)} = 0.1496$, and the true transshipment costs are $c_{13}^{(t)} = 0.0935$, $c_{14}^{(t)} = 0.1232$, $c_{23}^{(t)} = 0.1141$, $c_{25}^{(t)} = 0.0320$, $c_{34}^{(t)} = 0.1615$, and $c_{35}^{(t)} = 0.0867$.

Figure D.2. (Color online) Convergence of Average Cumulative Objective Function Errors
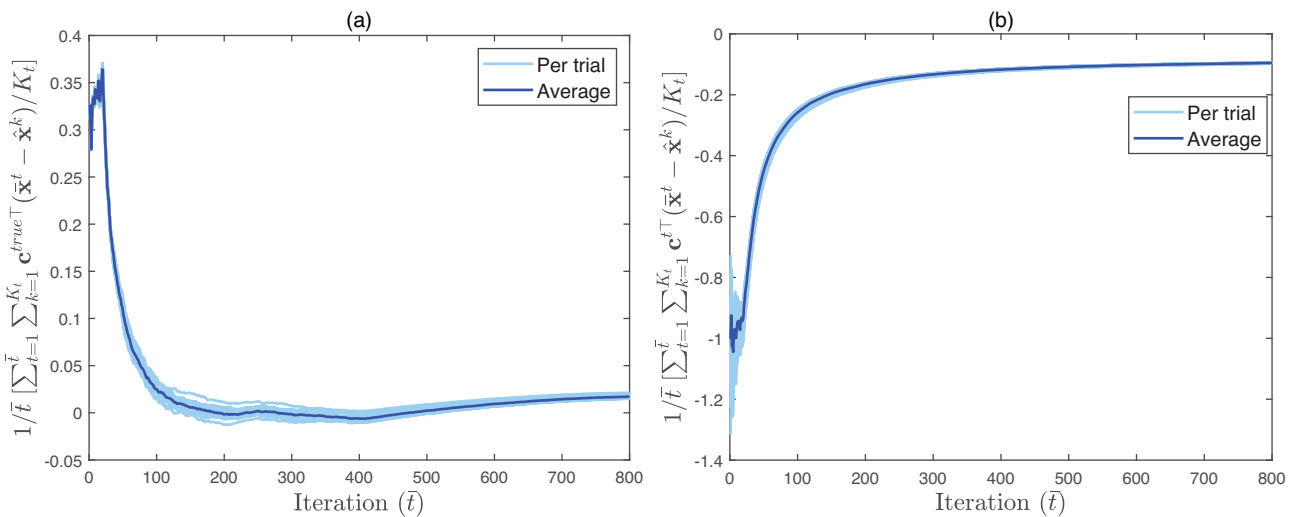
Notes. (a) Average cumulative error under $\mathbf{c}^{true}$. (b) Average cumulative error under $\mathbf{c}^t \in \mathcal{C}_t$.

References

Abbeel P, Ng AY (2004) Apprenticeship learning via inverse reinforcement learning. *Proc. 21st Internat. Conf. Machine Learn.* (Association for Computing Machinery, New York), 1–8.

Ahuja RK, Orlin JB (2001) Inverse optimization. *Oper. Res.* 49(5):771–783.

Aswani A, Shen ZJ, Siddiq A (2018) Inverse optimization with noisy data. *Oper. Res.* 66(3):870–892.

Aswani A, Shen ZJM, Siddiq A (2019) Data-driven incentive design in the Medicare Shared Savings Program. *Oper. Res.* 67(4):1002–1026.

Babier A, Chan TC, Lee T, Mahmood R, Terekhov D (2021) An ensemble learning framework for model fitting and evaluation in inverse linear optimization. *INFORMS J. Optim.* 3(2):119–138.

Bärmann A, Pokutta S, Schneider O (2017) Emulating the expert: Inverse optimization through online learning. Precup D, Teh YW, eds. *Proc. 34th Internat. Conf. Machine Learn.*, vol. 70 (PMLR), 400–410.

Bertsimas D, Mazumder R (2014) Least quantile regression via modern optimization. *Ann. Statist.* 42(6):2494–2525.

Bertsimas D, Gupta V, Paschalidis IC (2012) Inverse optimization: A new perspective on the Black-Litterman model. *Oper. Res.* 60(6):1389–1403.

Bertsimas D, Gupta V, Paschalidis IC (2015) Data-driven estimation in equilibrium using inverse optimization. *Math. Programming* 153(2):595–633.

Birge JR, Hortaçsu A, Pavlin JM (2017) Inverse optimization for the recovery of market structure from market outcomes: An application to the MISO electricity market. *Oper. Res.* 65(4):837–855.

Boutilier JJ, Lee T, Craig T, Sharpe MB, Chan TC (2015) Models for predicting objective function weights in prostate cancer IMRT. *Medical Phys.* 42(4):1586–1595.

Candes EJ, Wakin MB, Boyd SP (2008) Enhancing sparsity by reweighted ℓ_1 minimization. *J. Fourier Anal. Appl.* 14(5–6):877–905.

Chan TC, Lee T (2018) Trade-off preservation in inverse multiobjective convex optimization. *Eur. J. Oper. Res.* 270(1):25–39.

Chan TC, Lee T, Terekhov D (2019) Inverse optimization: Closed-form solutions, geometry and goodness of fit. *Management Sci.* 65(3):680–695.

Chan TC, Craig T, Lee T, Sharpe MB (2014) Generalized inverse multiobjective optimization with application to cancer therapy. *Oper. Res.* 62(3):680–695.

Darmon N, Ferguson EL, Briend A (2006) Impact of a cost constraint on nutritionally adequate food choices for French women: An analysis by linear programming. *J. Nutrition Ed. Behav.* 38(2):82–90.

Dawande M, Keskinocak P, Swaminathan JM, Tayur S (2001) On bipartite and multipartite clique problems. *J. Algorithms* 41(2):388–403.

Dong C, Chen Y, Zeng B (2018) Generalized inverse optimization through online learning. *Proc. 32nd Conf. Neural Inform. Processing Systems*, 86–95.

Erkin Z, Bailey MD, Maillart LM, Schaefer AJ, Roberts MS (2010) Eliciting patients' revealed preferences: An inverse Markov decision process approach. *Decision Anal.* 7(4):358–365.

Esfahani PM, Shafieezadeh-Abadeh S, Hanasusanto GA, Kuhn D (2018) Data-driven inverse optimization with imperfect information. *Math. Programming* 167(1):191–234.

Ghate A (2015) Inverse optimization in countably infinite linear programs. *Oper. Res. Lett.* 43(3):231–235.

Ghobadi K, Lee T, Mahmoudzadeh H, Terekhov D (2018) Robust inverse optimization. *Oper. Res. Lett.* 46(3):339–344.

Gurobi Optimization LLC (2020) Gurobi optimizer reference manual. Accessed April 13, 2021, http://www.gurobi.com.

Heuberger C (2004) Inverse combinatorial optimization: A survey on problems, methods, and results. *J. Combin. Optim.* 8(3):329–361.

Hubert M, Rousseeuw PJ, Van Aelst S (2008) High-breakdown robust multivariate methods. *Statist. Sci.* 23(1):92–119.

Iyengar G, Kang W (2005) Inverse conic programming with applications. *Oper. Res. Lett.* 33(3):319–330.

Keshavarz A, Wang Y, Boyd S (2011) Imputing a convex objective function. *2011 IEEE Internat. Sympos. Intelligent Control* (IEEE), 613–619.

Koenker R, Hallock KF (2001) Quantile regression. *J. Econom. Perspect.* 15(4):143–156.

Lamperski JB, Schaefer AJ (2015) A polyhedral characterization of the inverse-feasible region of a mixed-integer program. *Oper. Res. Lett.* 43(6):575–578.

Lee T, Hammad M, Chan TC, Craig T, Sharpe MB (2013) Predicting objective function weights from patient anatomy in prostate IMRT treatment planning. *Medical Phys.* 40(12):121706.

Mangasarian O (1979) Uniqueness of solution in linear programming. *Linear Algebra Appl.* 25(1):151–162.

Naghavi M, Foroughi AA, Zarepisheh M (2019) Inverse optimization for multi-objective linear programming. *Optim. Lett.* 13(2):281–294.

Perignon M, Masset G, Ferrari G, Barré T, Vieux F, Maillot M, Amiot MJ, Darmon N (2016) How low can dietary greenhouse gas emissions be reduced without impairing nutritional adequacy, affordability and acceptability of the diet? A modelling study to guide sustainable food choices. *Public Health Nutrition* 19(14):2662–2674.

Rousseeuw PJ (1984) Least median of squares regression. *J. Amer. Statist. Assoc.* 79(388):871–880.

Rousseeuw PJ, Van Driessen K (2006) Computing LTS regression for large data sets. *Data Mining Knowledge Discovery* 12(1):29–45.

Saez-Gallego J, Morales JM (2017) Short-term forecasting of price-responsive loads using inverse optimization. *IEEE Trans. Smart Grid* 9(5):4805–4814.

Schaefer AJ (2009) Inverse integer programming. *Optim. Lett.* 3(4):483–489.

Tavaslioǧlu O, Lee T, Valeva S, Schaefer AJ (2018) On the structure of the inverse-feasible region of a linear program. *Oper. Res. Lett.* 46(1):147–152.

Troutt MD, Brandyberry AA, Sohn C, Tadisina SK (2008) Linear programming system identification: The general nonnegative parameters case. *Eur. J. Oper. Res.* 185(1):63–75.

Wang L (2009) Cutting plane algorithms for the inverse mixed integer linear programming problem. *Oper. Res. Lett.* 37(2):114–116.

Xavier EC (2012) A note on a maximum k-subset intersection problem. *Inform. Processing Lett.* 112(12):471–472.

Yohai VJ (1987) High breakdown-point and high efficiency robust estimates for regression. *Ann. Statist.* 15(2):642–656.

Zahed Shahmoradi is a PhD candidate in the department of industrial engineering at the University of Houston. His research interests include topics at the intersection of data-driven decision making, inverse optimization, and data analytics.

Taewoo Lee is an assistant professor in the department of industrial engineering at the University of Houston. His primary research interests include inverse optimization and optimization under uncertainty and the application of optimization methods to healthcare operations and medical decision-making problems.