You Are What You Eat: A Preference-Aware Inverse Optimization Approach

Farzin Ahmadi* Tinglong Dai† Kimia Ghobadi*

*Department of Civil and Systems Engineering and Malone Center for Engineering in Healthcare, Johns Hopkins University, Baltimore, Maryland 21218, {fahmadi1,kimia}@jhu.edu

†Carey Business School, Johns Hopkins University, Baltimore, Maryland 21202, dai@jhu.edu

A key challenge in the emerging field of precision nutrition entails providing diet recommendations that reflect both the (often unknown) dietary preferences of different patient groups and known dietary constraints specified by human experts. Motivated by this challenge, we develop a preference-aware constrained-inference approach in which the objective function of an optimization problem is not pre-specified and can differ across various segments. Among existing methods, clustering models from machine learning are not naturally suited for recovering the constrained optimization problems, whereas constrained inference models such as inverse optimization do not explicitly address non-homogeneity in given datasets. By harnessing the strengths of both clustering and inverse optimization techniques, we develop a novel approach that recovers the utility functions of a constrained optimization process across clusters while providing optimal diet recommendations as cluster representatives. Using a dataset of patients' daily food intakes, we show how our approach generalizes stand-alone clustering and inverse optimization approaches in terms of adherence to dietary guidelines and partitioning observations, respectively. The approach makes diet recommendations by incorporating both patient preferences and expert recommendations for healthier diets, leading to *structural* improvements in both patient partitioning and nutritional recommendations for each cluster. An appealing feature of our method is its ability to consider *infeasible* but informative observations for a given set of dietary constraints. The resulting recommendations correspond to a broader range of dietary options, even when they limit unhealthy choices.

Key words: Inverse optimization, human-algorithm connection, diet recommendation, clustering

1. Introduction

Precision nutrition, an emerging field that addresses "the practical question of what to eat to stay healthy," has garnered significant attention in recent years, as imbalanced nutrition has emerged as a key contributor to numerous health issues that cost "hundreds of billions of dollars" each year (Rodgers and Collins 2020, p. 735). In relation to this question, adhering to healthy dietary habits plays an essential role. In practice, physicians often recommend specific diet regimens, for instance, the Dietary Approaches to Stop Hypertension (DASH) diet to control patients' sodium intake (DASH 2018). However, these diet regimens are rarely tailored to patients with their lifestyles

or dietary preferences in mind, making long-term patient adherence challenging (Bazrafkan et al. 2021, Downer et al. 2016, Inelmen et al. 2005). Incorporating dietary preferences helps generate more palatable diet recommendations (i.e., preference-aware diet recommendations), which is a crucial building block of precision nutrition (Rodgers and Collins 2020). Yet, dietary preferences, which are commonly reflected in each segment's objective functions, are rarely known; hence, broad and general dietary guidelines continue to be the norm.

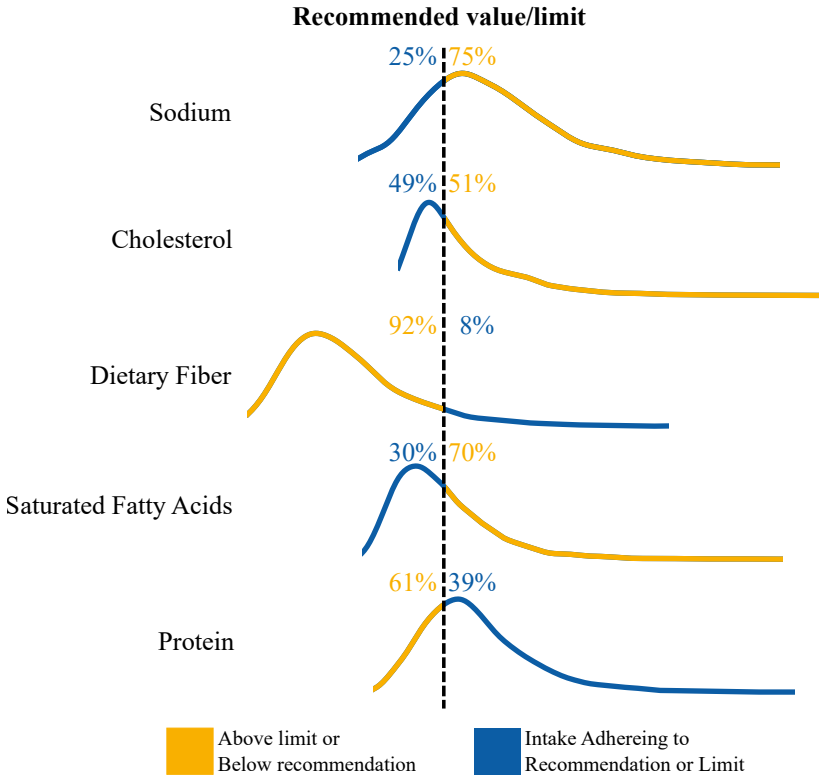


Figure 1 **A schematic figure representing the distribution of the nutritional intakes of the respondents from the NHANES dataset compared with the recommended values or limits set by the DASH dietary behavior plan. The majority of the respondents (at least half) exhibit unhealthy behaviors with regard to many essential nutrients, including sodium, cholesterol, dietary fiber, and saturated fatty acids.**

Interestingly, diet planning was among the pioneering application areas of operations research techniques, particularly linear programming (Dantzig 1965, Stigler 1945). In recent years, the focus of this area has shifted to using observed food-intake *data* to help inform diet recommendations. However, commonly used data-driven approaches are more suitable for replicating patient behaviors (both desirable and non-desirable ones), which are often at odds with expert recommendations. As an illustrative example, in Figure 1, we plot the dietary behaviors of the National Health and Nutrition Examination Survey (NHANES) respondents, which includes nearly 10,000 respondents

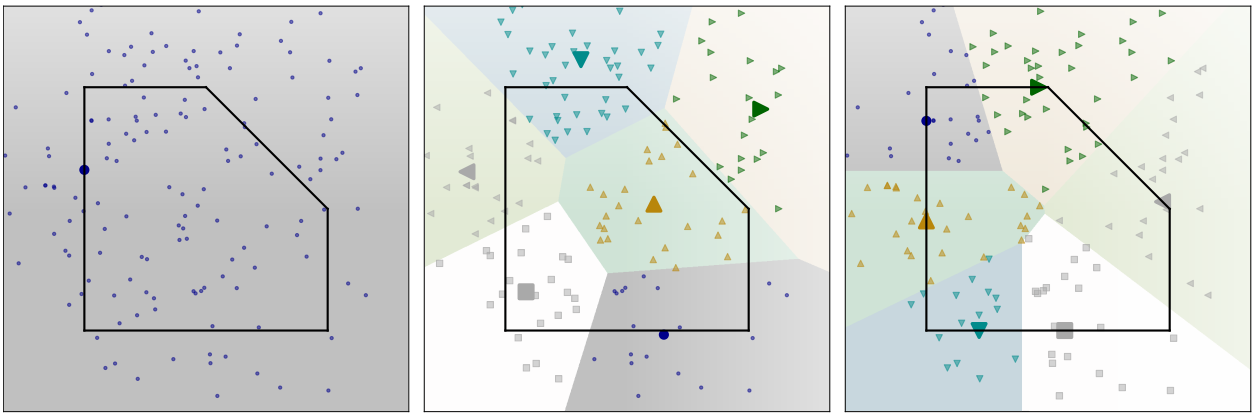
who record their daily food choices for two days.[1] As the figure shows, the majority of the respondents do not adhere to the limits set forth by the DASH dietary guidelines for a majority of categories. Clearly, a naïve clustering model can replicate the unhealthy behaviors that a significant portion of the patients exhibit. The presence of diverse patient objectives (e.g., regulating calories, maximizing taste, and minimizing cost) and multiple dietary constraints (e.g., fiber intake should be within a certain range) further complicates the diet recommendation problem. In such circumstances, human intervention is essential to guide diet recommendation algorithms.

Existing machine learning algorithms are not immediately suitable for this type of problem. As powerful as these algorithms are, when used to "recover" optimization problems, they mostly operate under the assumption that the optimization problems are not subject to hard constraints and handle occasional constraints by introducing large penalties.[2] Parallel to machine learning, inverse optimization focuses on recovering an optimization problem from a given set of observations (see, e.g., Ahuja and Orlin 2001, Aswani et al. 2018, Chan et al. 2019, Ghobadi et al. 2018); theoretically, the original optimization (a.k.a. "forward" optimization) problem is recovered such that the observation becomes its optimal solution. However, a shortcoming of inverse optimization is that it usually applies to a single or a small number of observations (Ahuja and Orlin 2001). Although applying inverse optimization to the average or the worst case of multiple observations is possible (Esfahani et al. 2018, Ghobadi et al. 2018), existing inverse optimization techniques assume all observations share the same feasible set and objective function.

Figure 2 illustrates a bi-dimensional constrained environment where decisions are observed over potentially different objectives while a proposed shared feasible set is in place. As Figure 2(a) shows, when inverse optimization models are applied to recover objective vectors and optimal decisions, they generally assume all observations share the same feasible set and objective function. A stand-alone inverse optimization model is not developed with partitioning in mind; thus, in the face of all the observed decisions, stand-alone inverse optimization will yield a single optimal solution pertaining to a one-size-fits-all scenario. Figure 2(b), on the other hand, shows that stand-alone clustering models do not consider the known constraints of the setting and result in suboptimal or even infeasible solutions. In other words, a naïve machine learning model is unable to capture the deviations and often provides recommendations that are—although similar to observed behaviors—undesirable (e.g., they tend to recommend diets high in sodium when patients' dietary choices tend to be excessively high in sodium). This tendency is in contrast to a combined model that

[1] https://www.cdc.gov/nchs/nhanes/index.htm

[2] One approach to address this major limitation of machine learning algorithms is "constraint learning," that is, learning constraints from examples. Yet, constraint learning is a nascent field in machine learning, with limited evidence of its effectiveness and few real-world applications (De Raedt et al. 2018).



a) Standalone Inverse Optimization

b) Standalone Clustering

c) Unifying Inverse Optimization and Clustering

Figure 2 **A representative example of how inverse optimization combined with machine learning incorporates expert recommendation in clustering non-homogeneous observations, whereas both stand-alone inverse optimization and machine learning overlook important aspects, namely the differences between observed decisions and the known constraints of the problem, respectively. In the figure, smaller, similar shapes showcase decisions within the same cluster, whereas larger shapes show the cluster representative that the model learns. The shaded areas represent the partitioning.**

incorporates the ability of machine learning models to capture non-homogeneity (i.e., potential variation across decision makers) and the ability of inverse optimization models to recover objective functions, which would provide better recommendations for each cluster of patients.

Most notably, such a combined approach produces *clusters* that differ from those produced by a stand-alone machine learning model in that each cluster contains a portion of the feasibility region with extreme points (which can be optimal for a realization of the unknown parameters of the optimization problem), whereas the stand-alone machine learning approach produces several interior clusters that are "landlocked" by other clusters. Figure 2(c) demonstrates how our approach yields *structurally* different solutions in terms of both partitioning and the recommended solutions for each cluster. Clearly, the combined approach is more than just layering an optimization problem on top of a clustering problem; rather, it allows for interaction between optimization and clustering.

In line with the spirit of Figure 2(c), we develop a novel methodology that unifies machine learning and inverse optimization (denoted as MLIO) for constrained decision-making environments with unknown performance metrics and known constraints. Our methodology leverages the strengths of clustering models (in handling large and non-homogeneous datasets) and inverse optimization (in recovering the utility function under constraints and providing optimality guarantees) in large datasets. Using this methodology, we enforce desirable constraints on the cluster representatives

and guarantee optimality conditions. Specifically, we formulate an optimization model that incorporates the known parameters of the constrained (forward) optimization problems and the set of all observed decisions. The model then optimizes over a loss criterion and recovers a number of clusters and the same number of (forward) optimization problems. We develop the model in such a way that for each of the recovered optimization problems and clusters, an optimal solution minimizing the loss towards the observed decisions within the cluster is also obtained. Furthermore, we present two solution approaches to the MLIO problem for larger datasets. The first, *sequential* approach treats the clustering and inverse optimization steps sequentially to provide a feasible solution for the general problem, whereas the second, *embedded* approach builds on an initial partitioning by solving the inverse optimization model and reassigns observations jointly and iteratively.

Additionally, we demonstrate that for the case of infeasible observed solutions, our proposed methodology has the desirable effect of providing optimal solutions that deviate incrementally from existing solutions, as shown in Figure 2(c) for clusters including infeasible solutions. For our application, we specifically focus on the personalized diet recommendation problem. We show the performance and applicability of these solution approaches to the setting of the diet recommendation problem for a subset of patients from the NHANES dietary data. We show a naïve clustering model results in the replication of patients' existing dietary patterns that may reflect unhealthy lifestyles. By contrast, our MLIO model, through added flexibility of human inputs in the form of dietary constraints, recommends diets that adhere to DASH nutritional guidelines while incorporating dietary preferences inferred from the data. Even though they restrict unhealthy options, the resulting recommendations often correspond to a wider variety of dietary options.

The rest of this paper is organized as follows. In Section 2, we review the related literature. Next, we describe the problem setting from a theoretical standpoint in Section 3 and detail our general MLIO model that unifies machine learning and inverse optimization. Section 4 discusses the solution method, where Sections 4.1 and 4.2 detail and compare two efficient solution methods. Finally, Section 5 applies our novel approach to a diet recommendation problem and generates managerial implications. This paper concludes in Section 6.

2. Literature

Our paper builds on and contributes to several streams of literature. Methodology-wise, we blend inverse optimization and unsupervised machine learning, so our paper is related to the literature on both approaches. Thematically, we contribute to the emerging field of precision nutrition and extend the scope of the healthcare operations management literature. More broadly, our paper is connected to the nascent literature on the human-algorithm connection.

Inverse optimization is an inference tool for recovering optimization models and has received increased attention since the seminal work by Ahuja and Orlin (2001), who develop a method to

recover the cost vector of a linear programming problem given the optimal solution of the problem. Interest in inverse optimization first arose from finding parameters for combinatorial optimization problems such as the shortest-path problem from given optimal solutions (Burton and Toint 1992, Zhang and Ma 1999). In general, inverse optimization models aim to recover unknown parameters of an optimization problem by minimizing the loss in the optimality of the observed solutions. Early work in inverse optimization focuses on the case of a *single* observed solution (see, e.g., Ahuja and Orlin 2001, Chan et al. 2019, Iyengar and Kang 2005, Schaefer 2009). To overcome this limitation, several recent papers (Aswani et al. 2018, Babier et al. 2021, Ghobadi and Mahmoudzadeh 2020, Shahmoradi and Lee 2022b) have extended this methodology to a limited number of (possibly noisy and/or suboptimal) observations. Recent inverse optimization works have shown the applicability of this approach in various health and non-health applications (Ahmadi et al. 2020a, Akhtar et al. 2022, Aswani et al. 2019, Bärmann et al. 2018, Beil and Wein 2003, Bertsimas et al. 2012, Chan et al. 2014, 2022a,b, Chow and Recker 2012, Faragó et al. 2003, Shahmoradi and Lee 2022a,b); we refer the reader to Chan et al. (2021) for a comprehensive survey of recent advances in theory and applications of inverse optimization. However, an underlying assumption across all these studies is that the observed decisions are solutions to the same optimization problem. In other words, all of the observed solutions are derived from the same utility function and the same feasible set. To the best of our knowledge, existing inverse optimization methods are not capable of distinguishing between observations that share the same feasible set but potentially different objectives and provide no means of partitioning given observations. This paper bridges this gap in the literature.

Various machine learning applications are also considered for constrained optimization problems. From this perspective, the majority of the literature focuses on methods to solve optimization problems (especially combinatorial optimization problems). The reader is referred to Bengio et al. (2021) for a review of machine learning applications to continuous and combinatorial optimization problems. However, only a handful of studies have focused on learning solutions to optimization problems from given decisions. Among these studies, some map existing, noisy, and/or suboptimal observed decisions to better, near-optimal solutions based on the available knowledge of the problem context (Misra et al. 2018). Deep neural network and reinforcement learning models have also been explored to generate "good" solutions for learning problems that involve decision making (Bengio et al. 2021, Bastani et al. 2021). However, most of the existing machine learning-based models do not guarantee the optimality of the mapped solutions and are prone to missing or violating important constraints in the optimization problem (Misra et al. 2018), with a few exceptions: Márquez-Neila et al. (2017), for example, explore the idea of imposing hard constraints on deep networks to guarantee optimality; as another example, Elmachtoub and Grigas (2022) propose alternative loss functions in the prediction models that account for subsequent optimization problems. Machine

learning techniques have also been used to infer the objective weights of optimization problems related to medical decision-making (Babier et al. 2018, Beam and Kohane 2018). Numerous studies investigate machine learning approaches and applications in healthcare services, for reviews of which we refer the reader to Firdaus et al. (2018) and Waring et al. (2020). Overall and mainly due to limitations from machine learning models to satisfy optimality of learned solutions in constrained data-driven settings, these models are not widely used for inference in constrained environments.

Taken together, neither machine learning (due to a general lack of optimality guarantees) nor inverse optimization (due to the inability to handle observations from different problems) alone is ideal for data-intensive constrained inference. However, by leveraging the strengths of both approaches, we develop a novel method that provides meaningful inference of unknown parameters and learns optimal solutions. The main contribution of our method is in its ability to recover different values for the unknown parameters of a general optimization problem through clustering a set of observations. We do so by using an augmented machine learning approach embedded with inverse optimization techniques that allow for optimally partitioning such observations. We show such an approach is superior to stand-alone applications of machine learning models in providing optimality guarantees for recommendations.

Our paper additionally contributes to the emerging stream of literature on human-algorithm connections. Dietvorst et al. (2015) define algorithm aversion as the tendency of forecasters and decision makers to use a human forecaster or algorithm against a statistical or evidence-based model, even when the evidence-based algorithms outperform human models. They also suggest that allowing users to modify algorithms mitigates algorithm aversion. This effect has been further examined in more recent research through incorporating considerations such as partial adherence to recommendations (Grand-Clément and Pauphilet 2022), treatment adherence (Lin et al. 2022), patients' resistance to medical artificial intelligence (Longoni et al. 2019), physicians' reputation concerns (Dai and Singh 2020), experts' updated belief in the algorithm's accuracy (de Verícourt and Gurkan 2022), and augmentation of algorithmic decisions with human knowledge (Chen et al. 2022). Our paper is consistent with the literature in that we incorporate individuals' (unknown) dietary preferences when making diet recommendation decisions. Unlike previous research, which considers *unconstrained* decision environments, our framework allows the algorithm's recommendations to explicitly reflect known decision constraints.

Our paper was motivated by a diet recommendation problem, which is among the earliest application areas of operations research. The seminal work by Stigler (1945) models diet recommendation as an optimization problem and triggers broad interest among the optimization community (Dantzig 1965). The original diet recommendation problem entails finding the optimal intake amounts of different food items based on given constraints on food types and nutrients and given

cost functions. Later studies have considered the diet recommendation problem at the individual (Maillot et al. 2010) and community levels (Buttriss et al. 2014, Morgenstern et al. 2021). Other studies in the literature point out influential factors (e.g., convenience and taste) in dietary choices and note that recommendations should be focused on such aspects (Irz et al. 2016). Optimization models have been used to model the diet recommendation problem as well (Gazan et al. 2018, Ghobadi et al. 2018). Gazan et al. (2018) provide a review of diet optimization models that take both sustainability and acceptability into account. However, diet models usually suffer from recommendations that are not personalized for the population in question due to inadequate access to suitable cost functions or ill-defined feasible sets that are either infeasible or do not reflect the desirable features that the patients demand, or due to sole reliance on the observed behaviors in machine learning models that hinder adherence to healthier recommendations. Both inverse optimization models (Ahmadi et al. 2020a, Ghobadi et al. 2018) and machine learning approaches (Ivancic et al. 2020) have considered the problem of providing meaningful and acceptable diets based on the observed decisions of patients. Although recent works consider providing clustering results with optimality guarantees for the preferences (Shahmoradi and Lee 2022a) and consider different objective vectors for each observation (Birge et al. 2022), to the best of our knowledge, our paper is the first to develop a hybrid approach to jointly (1) clustering individuals based on their dietary behaviors and (2) proposing optimal solutions and recovering unknown optimization parameters for each group.

Our work also contributes to the vibrant healthcare operations management literature, in which more tailored diet recommendations can be viewed as a gateway to improved quality of care. The literature in healthcare operations management is broad and ever-evolving. We refer the reader to Dai and Tayur (2020), Keskinocak and Savva (2020), and Terwiesch et al. (2020) for reviews of the healthcare operation management literature. Evidence within the healthcare operations management literature shows human judgment and expert recommendation could improve prediction' accuracy (Dai and Singh 2022, Ibrahim et al. 2021), and non-personalized treatment schedules are not optimal for different groups of patients (Suen et al. 2022).

3. A Preference-Aware Inverse Optimization Approach

We consider a constrained decision-making environment where each decision maker (e.g., patient) is subject to a fixed and known feasible set. Different decision makers may have diverse objective functions even with shared decision constraints. Given a collection of observations, we develop a method that optimally partitions the observed decisions and recovers cost vectors in such a way that the representative decisions in each cluster are optimal for the recovered cost vector. This problem has parallels in applied settings such as the daily dietary behaviors of individuals who

follow a particular diet (e.g., the DASH diet) and existing radiation therapy treatment plans for different patients (Goldenberg et al. 2019). An appealing feature of our method is its ability to consider *infeasible* observations for the given feasible set. Such observations may contain important information as the users may not be able or choose not to fully satisfy all constraints. For instance, daily dietary behavior contains information about the user's preferences and palate, regardless of its feasibility for a particular diet.

We consider a set of different observations over a fixed and known polyhedral feasible set and assume the utility functions of the decision makers for the decisions are linear but unknown. The linearity assumption allows for a more tractable formulation while also allowing the objective function to be monotonous in terms of dietary preferences. Our approach consists of partitioning and learning: the partitioning component generates clusters of homogeneous decisions, and the learning component learns cluster representatives that optimally recover the unknown objective function based on observations within the cluster.

In the rest of this section, we first lay the groundwork of our approaches in Section 3.1 by setting up the modeling environment and specifying the learning component. Section 3.2 presents the general problem of combing machine learning and inverse optimization. Section 3.3 formulates the general problem as a mixed-integer bilinear program.

3.1. Preliminary: Inverse Optimization

We consider a setting where a set of potentially non-homogeneous observations is given along with a feasible set shared by all the observations. Corresponding to this setting is a forward linear optimization model FO, where $\mathbf{c} \in \mathbb{R}^n$ is the cost vector forming the utility function and $\Omega \subseteq \mathbb{R}^n$ is the feasible set where $\Omega \neq \emptyset$:

$$\text{FO}(\mathbf{c}, \Omega): \underset{\mathbf{x}}{\text{maximize}} \quad \mathbf{c}'\mathbf{x} \tag{1a}$$

$$\text{subject to} \quad \mathbf{A}\mathbf{x} \geq \mathbf{b}, \tag{1b}$$

$$\mathbf{x} \in \mathbb{R}^n. \tag{1c}$$

In the above formulation, $\mathbf{x} \in \mathbb{R}^n$ represents the decision variables of FO. The matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ and the vector $\mathbf{b} \in \mathbb{R}^m$ (m is the number of constraints) constitute the required parameters to define the fixed feasible set $\Omega \subseteq \mathbb{R}^n$ of the forward optimization problem, and we have $\Omega = \{x \in \mathbb{R}^n \mid \mathbf{A}\mathbf{x} \geq \mathbf{b}\}$. Note the optimal solution set $\mathbf{X}^* \subseteq \Omega$ for FO depends on the value of $\mathbf{c}$. For a polyhedral feasible set Ω, we denote the boundary of Ω as Ω^{opt}, which contains all points in Ω that can be optimal for some cost vector $\mathbf{c} \neq \mathbf{0}$. For a non-homogeneous set of observed decisions $\mathbf{X} \subseteq \mathbb{R}^n$, our goal is to partition the given set of observed decisions into a given number of clusters L such that the partitioning decision reflects the feasible set Ω and each cluster pertains to a group of decision-makers with

the same objective function. Whereas clustering algorithms provide optimal partitioning (based on some metric) of the observations in the absence of constraints, we aim to incorporate the additional knowledge of the constraints into the partitioning scheme. To that end, we first discuss the inverse learning model as the specific inverse optimization approach we use to recover unknown cost vectors and learn optimal solutions to linear optimization problems. Then, we combine clustering and inverse optimization to recover optimization models for non-homogeneous observed decisions.

The learning component of our approach corresponds to an inverse optimization model that is capable of recovering unknown parameters of optimization problems given homogeneous observed decisions. Because our goal is to generate clusters and cluster representatives that are optimal for the recovered utility functions, our inverse optimization model is the inverse learning framework, which is capable of learning optimal solutions. The core idea of inverse learning is to find a solution $\mathbf{z} \in \mathbb{R}^n$ that results from minimal perturbation of given observed decisions $\mathbf{X}_0$ attributed to FO to a singular optimal solution. By its nature of being an inverse optimization model, an inverse learning model is also capable of recovering a cost vector $\mathbf{c}$ that makes the perturbed solution $\mathbf{z}$ optimal for FO$(\mathbf{c}, \Omega)$. We denote by $\mathbf{y} \in \mathbb{R}^m$ the dual variables associated with constraints forming Ω; $\mathbf{z} \in \mathbb{R}^n$ the perturbed solution contained on Ω^{opt}; $\mathbf{E} \in \mathbb{R}^{n \times K}$ the perturbation matrix for the observed decisions; and ϵ^k be the k^{th} column of $\mathbf{E}$. Then, the inverse learning model is as follows:

$$\text{IO}(\mathbf{X}_0, \Omega): \underset{\mathbf{c},\mathbf{y},\mathbf{z},\mathbf{E}}{\text{minimize}} \quad \mathcal{D}(\mathbf{z}, \mathbf{X}_0) \tag{2a}$$

$$\text{subject to} \quad \mathbf{A}\mathbf{z} \geq \mathbf{b}, \tag{2b}$$

$$\mathbf{c}'\mathbf{z} = \mathbf{b}'\mathbf{y}, \tag{2c}$$

$$\mathbf{A}'\mathbf{y} = \mathbf{c}, \tag{2d}$$

$$\mathbf{z} = (\mathbf{x}^k - \epsilon^k), \quad \forall k \in \mathcal{K} \tag{2e}$$

$$\sum_{j=1}^{m} y_j = 1, \tag{2f}$$

$$\mathbf{y} \geq \mathbf{0}. \tag{2g}$$

In the inverse learning model, the objective is to minimize some metric $\mathcal{D}$ between the learned solution $\mathbf{z}$ and the given observed decisions $\mathbf{X}_0$. The constraints of IO guarantee the feasibility (2b) and optimality (2c) of $\mathbf{z}$ for FO$(\mathbf{c}, \Omega)$. Note that although IO is non-convex, a growing body of literature proposes methods to solve IO by re-formulating it as a series of convex models (Aswani et al. 2018, Esfahani et al. 2018, Ghobadi and Mahmoudzadeh 2020). Additionally, it has been shown that IO can be remodeled as a linearly constrained and convex optimization problem under mild assumptions on the constraint matrix $\mathbf{A}$ (Ahmadi et al. 2020a, Chan et al. 2019). Ahmadi et al. (2020a) show that when $\Omega \neq \emptyset$, IO is feasible for any $\mathbf{X}_0$ and for any metric $\mathcal{D}$. We use

this result in later sections to formulate a general model for non-homogeneous decisions $\mathbf{X}$ that is always feasible when $\Omega \neq \emptyset$.

Although IO is a powerful tool for recovering utility functions from observed decisions, it has an underlying assumption that $\mathbf{c}$ is the same as all observed decisions and that $\mathbf{X}_0$ is homogeneous. As such, by definition, IO is not capable of handling non-homogeneity in the observed decisions. In what follows, we consider clustering methodologies and develop a combined approach for learning a series of cost vectors and partitioning observed decisions.

3.2. Recovering Optimization Models from Non-homogeneous Datasets

In the presence of a non-homogeneous set of observed decisions, $\mathbf{X}$, over multiple unknown utility functions $\mathbf{c}_1, \ldots, \mathbf{c}_L$, the practicality of the IO model discussed in the previous section is limited because IO is capable of learning a unified cost vector for all observed decisions. The presence of non-homogeneous decisions is analogous to a set of observed decisions from multiple decision makers/users over the same feasible set (e.g., the diet recommendation problem for different patient groups over the same dietary requirements). We aim to provide a method that can optimally (by minimizing a metric) partition the set of observed decisions by identifying the decisions that share the same decision maker and proposing a utility function for that decision maker. As such, we formulate a problem that simultaneously partitions the observed decisions into a given number of groups and recovers optimization models and their optimal solutions for each group.

Let $\mathbf{X} \subseteq \mathbb{R}^n$ be a set of potentially non-homogeneous decisions made over Ω. We aim to partition $\mathbf{X}$ into a given number of clusters $\mathbf{X}_1, \ldots, \mathbf{X}_L$ while the model recovers utility parameters $\mathbf{c}_1, \ldots, \mathbf{c}_L$ and generates optimal solutions $\mathbf{x}_1, \ldots, \mathbf{x}_L$ for the recovered optimization problems $\mathrm{FO}_1(\mathbf{c}_1, \Omega), \ldots, \mathrm{FO}_L(\mathbf{c}_L, \Omega)$ such that $\mathbf{x}_l$ are the optimal solutions contained in $\Omega^{opt}(\mathbf{c}_l) \subseteq \mathbb{R}^n$, the set of all optimal solutions to $\mathrm{FO}(\mathbf{c}_l, \Omega)$ $\forall l \in \mathcal{L} = \{1, \ldots, L\}$. We formulate this partitioning problem as follows:

$$
\begin{aligned}
\mathrm{MLIO}(\mathbf{X}, \Omega, L): \underset{\mathbf{X}_1,\ldots,\mathbf{X}_L,\mathbf{x}_1,\ldots,\mathbf{x}_L,\mathbf{c}_1,\ldots,\mathbf{c}_L}{\text{minimize}} \quad & \sum_{l=1}^{L} \mathcal{D}(\mathbf{x}_l, \mathbf{X}_l) && \text{(3a)}\\
\text{subject to} \quad & \mathbf{x}_l \in \Omega^{opt}(\mathbf{c}_l), \quad \forall l \in \mathcal{L} && \text{(3b)}\\
& \mathbf{X}_\lambda \cap \mathbf{X}_\gamma = \emptyset, \quad \forall \lambda, \gamma \in \mathcal{L}, \lambda \neq \gamma && \text{(3c)}\\
& \bigcup_{l=1}^{L} \mathbf{X}_l = \mathbf{X}, && \text{(3d)}\\
& \mathbf{c}_l = \mathbf{0} \quad \text{iff } \Omega = \mathbb{R}^n. \quad \forall l \in \mathcal{L} && \text{(3e)}
\end{aligned}
$$

We refer to formulation (3) as the machine learning and inverse optimization (MLIO) model. The MLIO model has four types of constraints: First, constraint (3b) ensures learned solutions $\mathbf{x}_l$ are optimal for $\mathrm{FO}(\mathbf{c}_l, \Omega)$. Next, constraints (3c) and (3d) impose the necessary criteria for

$\{\mathbf{X}_1, \ldots, \mathbf{X}_L\}$ to be a partition of $\mathbf{X}$. In particular, constraint (3c) ensures each observed decision will be included in exactly one cluster. Lastly, constraint (3e) ensures MLIO generalizes the unconstrained clustering problem. In the case of $\Omega = \mathbb{R}^n$, by setting all $\mathbf{c}_l = \mathbf{0}$, MLIO reduces to the unconstrained clustering problem, and the centroids that minimize $\mathcal{D}$ over $\mathbb{R}^n$ are achieved.

The objective of MLIO is to minimize the difference between the observed decisions in each cluster and the corresponding learned optimal solution for that cluster. In other words, MLIO looks for a partitioning of observed decisions that minimizes the total loss (as measured by the metric $\mathcal{D}$) for all clusters. We keep the metric $\mathcal{D}$ generic and allow it to be specified by the human expert. To show MLIO is well defined, we first outline the feasibility of MLIO for different values of L.

PROPOSITION 1. *MLIO*$(\mathbf{X}, \Omega, L)$ *is feasible for all* $L \in \mathbb{N}$.

Proposition 1 shows MLIO is well-defined for any given set of observed decisions; for $\Omega \neq \emptyset$, the feasible set of MLIO$(\mathbf{X}, \Omega, L)$, denoted by Ψ_L, is non-empty. An important characteristic of the MLIO model is that the feasible set of MLIO covers all possible partitions of $\mathbf{X}$, denoted by $\mathbf{P}_L(\mathbf{X})$. In other words, for any partition of $\mathbf{X}$ to L clusters, at least one solution for MLIO exists that results in that partition. This is formalized in Proposition 2.

PROPOSITION 2. *Let* $\hat{\Psi}_L$ *be the set of all partitions* $(\mathbf{X}_1, \ldots, \mathbf{X}_L) \in \mathbf{P}_L(\mathbf{X})$ *such that* $\exists$ $(\{\mathbf{x}_1, \ldots, \mathbf{x}_L\}, \{\mathbf{c}_1, \ldots, \mathbf{c}_L\})$, *where* $(\mathbf{X}_1, \ldots, \mathbf{X}_L, \mathbf{x}_1, \ldots, \mathbf{x}_L, \mathbf{c}_1, \ldots, \mathbf{c}_L) \in \Psi_L$. *Then,* $\hat{\Psi}_L$ *covers* $\mathbf{P}_L(\mathbf{X})$.

Propositions 1 and 2 establish that solving MLIO results in the optimal partition of $\mathbf{X}$ with the desirable feature of recovering the appropriate number of optimization problems $\text{FO}_1, \ldots, \text{FO}_L$ and their corresponding optimal solutions $\mathbf{x}_1, \ldots, \mathbf{x}_L$ and cost vectors $\mathbf{c}_1, \ldots, \mathbf{c}_L$. Furthermore, the definition of MLIO provides a generalized approach for bridging the unsupervised machine learning (clustering) problem and the multi-observation inverse optimization problem. For each cluster $\mathbf{X}_l$, MLIO recovers a cost vector $\mathbf{c}_l$ and a solution $\mathbf{x}_l$ that is rendered optimal for FO$(\mathbf{c}_l, \Omega)$, so setting $\Omega = \mathbb{R}^n$ reduces MLIO to the unsupervised learning problem where all recovered cost vectors are equal to the zero vector and the learned solutions are the ones that minimize the metric over $\mathbb{R}^n$. On the other hand, setting $L = 1$ reduces MLIO to the inverse learning problem, where all observed decisions are assumed to emerge from the same optimization problem and the same cost vector $\mathbf{c}$.

Table 1 compares (1) unsupervised machine learning approaches (K-means), (2) inverse optimization models (IO), and (3) the hybrid model (MLIO). As the table shows, the main advantage of considering MLIO over existing inverse optimization models is MLIO's capability to handle potentially non-homogeneous decisions. As such, MLIO provides a general clustering approach that is aware of the decisions that are feasible for the problem (Ω) where optimal clusters are learned

Table 1 Comparison between unsupervised machine learning models, inverse optimization models, and MLIO

Features	Unconstrained Clustering (e.g. K-means)	IO (e.g. Inverse learning)	MLIO
Proposing representative decisions	✓	✓	✓
Clustering capabilities	✓		✓
Considering non-trivial feasible sets ($\neq \mathbb{R}^n$)		✓	✓
Utility Function recovery		✓	✓
Learning optimal decisions		✓	✓

based on the given metric $\mathcal{D}$. The resulting optimal solution set of MLIO is a partition of the original observed decisions into L groups, each group representing decisions that the formulations recognize as being for the same optimization problem and sharing the same cost vector. MLIO additionally provides optimal solutions for each such group in the partition.

The definition provided in formulation (3) for MLIO is general and can be applied to any setting where inverse optimization can be modeled to learn optimal solutions. However, how it can be explicitly modeled for different classes of FO problems is not obvious. As such, in what follows, we provide an explicit formulation for solving MLIO as a mixed-integer bilinear optimization problem that finds an optimal solution to MLIO. We provide this formulation by building on the modeling approaches and techniques used for formulating IO.

3.3. A Mixed-Integer Bilinear Formulation

We now discuss how to explicitly model MLIO as an optimization problem and how to solve MLIO in a linear setting. MLIO can be modeled as a mixed-integer bilinear problem (MLIO-MIBP) in general by including binary variables indicating the inclusion of each observed decision in a cluster. We denote by $\mathbf{x}_l \in \mathbb{R}^n$ the learned solution for cluster $l \in \mathcal{L}$ and $\mathbf{v} \in \mathbb{R}^{K \times L}$ a binary matrix indicating the inclusion of observed decisions in each cluster, where $v_{k,l} = 1$ if and only if observed decision k is in cluster l. Note that a one-to-one correspondence exists between the matrix $\mathbf{v}$ and the partition in each feasible solution to MLIO. We then have the following formulation:

$$\begin{aligned}
\text{MLIO-MIBP}(\mathbf{X}, \Omega, L): \underset{\mathbf{x},\mathbf{v},\mathbf{Y},\mathbf{C},\mathbf{E}}{\text{minimize}} \quad & \sum_{l=1}^{L}\sum_{k=1}^{K} \mathcal{D}(\mathbf{x}_l, \mathbf{x}^k) \cdot v_{k,l} \\
\text{subject to} \quad & \mathbf{A}\mathbf{x}_l \geq \mathbf{b}, \quad \forall l \in \mathcal{L} && \text{(4a)} \\
& \mathbf{c}_l'\mathbf{x}_l = \mathbf{b}'\mathbf{y}_l, \quad \forall l \in \mathcal{L} && \text{(4b)} \\
& \mathbf{A}'\mathbf{y}_l = \mathbf{c}_l, \quad \forall l \in \mathcal{L} && \text{(4c)}
\end{aligned}$$

$$\mathbf{x}_l \cdot v_{k,l} = (\mathbf{x}^k - \epsilon^k) \cdot v_{k,l}, \quad \forall k \in \mathcal{K}, l \in \mathcal{L} \tag{4d}$$

$$\sum_{j=1}^{m} y_l^j = 1, \quad \forall l \in \mathcal{L} \tag{4e}$$

$$\sum_{l=1}^{L} v_{k,l} = 1, \quad \forall k \in \mathcal{K} \tag{4f}$$

$$\mathbf{y}_l \geq \mathbf{0}, \quad \forall l \in \mathcal{L} \tag{4g}$$

$$\mathbf{v}_{k,l} \in \{0,1\}, \quad \forall k \in \mathcal{K}, l \in \mathcal{L}. \tag{4h}$$

In the above formulation, constraints (4a), (4b), (4c), (4e), and (4g) are similar to their definitions in (2). Constraints (4d) ensure observed decisions sharing the same cluster are perturbed to a single solution. The constraints in formulation (4) ensure these perturbed solutions are contained in $\Omega^{opt}(\mathbf{c}_l)$ for each cluster and, hence, are optimal for their respective recovered cost vectors. Constraints (4f) ensure each observed decision is included in exactly one cluster. Note that for the special case of $\Omega = \mathbb{R}^n$, without loss of generality, we can remove constraints (4a), (4b), (4c), (4e), and (4g) because $\mathbf{A}$ and $\mathbf{b}$ do not exist. In this case, MLIO-MIBP also reduces to the general clustering problem for the metric $\mathcal{D}$ and L number of clusters. Finally, note MLIO-MIBP is developed as a generalization to the IO formulation to incorporate non-homogeneity in the observed decisions. Proposition 3 formalizes this statement.

PROPOSITION 3. *Let* $(\mathbf{x}^*, \mathbf{v}^*, \mathbf{Y}^*, \mathbf{C}^*, \mathbf{E}^*)$ *be optimal for MLIO-MIBP*$(\mathbf{X}, \Omega, L)$*. Then, for all* $l \in \mathcal{L}$, $(\mathbf{C}_l^*, \mathbf{Y}_l^*, \mathbf{x}_l^*, \hat{\mathbf{E}}_l^*)$ *is optimal for IO*$(\mathbf{X}_l^*, \Omega)$*, where* $\mathbf{x}_l^*, \mathbf{C}_l^*, \mathbf{Y}_l^*$ *are the* l^{th} *rows of* $\mathbf{x}^*, \mathbf{Y}^*, \mathbf{C}^*$ *and* $\hat{\mathbf{E}}_l^*$ *is the matrix of rows of matrix* $\mathbf{E}_l^*$ *for which* $v_{k,l} = 1$.

Proposition 3 shows how MLIO-MIBP generalizes IO for non-homogeneous decisions. Instead of learning only one solution and recovering only one cost vector, MLIO-MIBP is capable of learning L optimal solutions and recovering L cost vectors for each cluster. We note that although MLIO-MIBP is a mixed-integer bilinear program, it can be reformulated as a mixed-integer linear program (Gupte et al. 2013).

Next, in Theorem 1, we show the important result that MLIO-MIBP is equivalent to MLIO.

THEOREM 1. $(\mathbf{x}^*, \mathbf{v}^*, \mathbf{Y}^*, \mathbf{C}^*, \mathbf{E}^*)$ *is optimal for MLIO-MIBP*$(\mathbf{X}_0, \Omega, L)$ *if and only if* $\exists\ \bar{\mathbf{X}} = \{\mathbf{X}_1, \ldots, \mathbf{X}_L\} \in \mathbf{P}_L(\mathbf{X})$*, such that* $(\mathbf{X}, \mathbf{x}^*, \mathbf{C}^*)$ *is optimal for MLIO*$(\mathbf{X}, \Omega, L)$.

The MLIO-MIBP formulation detailed in this section provides an explicit formulation to solve MLIO and learn the optimal partition of observed decisions in a general, potentially constrained environment. However, recall from Section 3.1 that the presence of a combination of binary variables and bilinear or non-convex constraints in MLIO-MIBP makes this model computationally expensive in the general case. Thus, in the following section, we provide two computationally efficient solution approaches to MLIO.

4. Solution Approaches: Sequential vs. Embedded

In this section, we introduce two solution approaches to MLIO. Section 4.1 discusses a *sequential* solution approach where we first cluster the observed decisions independently from the given feasible set (equivalent to solving the unsupervised learning problem) and use IO to find optimal solutions and recover optimization problems for each cluster. Then, Section 4.2 discusses an iterative, *embedded* solution approach that finds clusters by minimizing the loss of points that are optimal for Ω from each cluster. The partition that is found by minimizing a convergence gap is then returned by the algorithm. Finally, Section 4.3 compares the two solution approaches in terms of optimal objective and run time with MLIO.

4.1. A Sequential Approach to Solving MLIO

Our first solution approach to tackling the complexity of MLIO-MIBP is to separate the clustering component from the learning component. In other words, we provide an approach that utilizes clustering models, such as the K-means method, to perform the clustering component of the learning process and use the IO model on each of the resulting clusters to learn optimal solutions. This approach provides an efficient method for finding a feasible solution to MLIO because inverse linear optimization models can be broken down into linearly constrained optimization problems. Figure 3 provides a schematic diagram of this approach. As shown in the figure, the non-homogeneous observed decisions $\mathbf{X}$ undergo a clustering scheme using an unsupervised machine learning model, and the resulting L clusters are each used as inputs to the IO model, alongside the known feasible set Ω one by one, resulting in recovering at most L forward optimization problems and learning at most L optimal solutions. We denote each learned optimal solution $\mathbf{x}_l$ as a sequential machine learning and inverse optimization (SEQ-MLIO) representative of cluster l. Whereas the SEQ-MLIO approach is computationally efficient, it bears the assumption that decisions are not influenced by the given feasible set and decisions closer to each other belong in the same cluster, which might not hold in highly constrained settings. Nevertheless, the SEQ-MLIO approach serves as a useful benchmark that can readily find a feasible solution for MLIO. Proposition 4 states that the resulting solution of the approach is a feasible solution to MLIO.

PROPOSITION 4. *The solution* $(\{\mathbf{X}_0^1,\ldots,\mathbf{X}_0^L\},\{\mathbf{x}_1^*,\ldots,\mathbf{x}_L^*\},\{\mathbf{c}_1,\ldots,\mathbf{c}_L\})$ *resulting from the SEQ-MLIO approach is a feasible solution for MLIO.*

Proposition 4 shows the solution obtained by SEQ-MLIO is feasible for MLIO. Yet, SEQ-MLIO does not provide a quality guarantee, because by design, it does not incorporate the available knowledge on Ω in the partitioning process, which plays a vital role in determining similarities among decisions. Furthermore, under SEQ-MLIO, the resulting partitioning suffers from the same

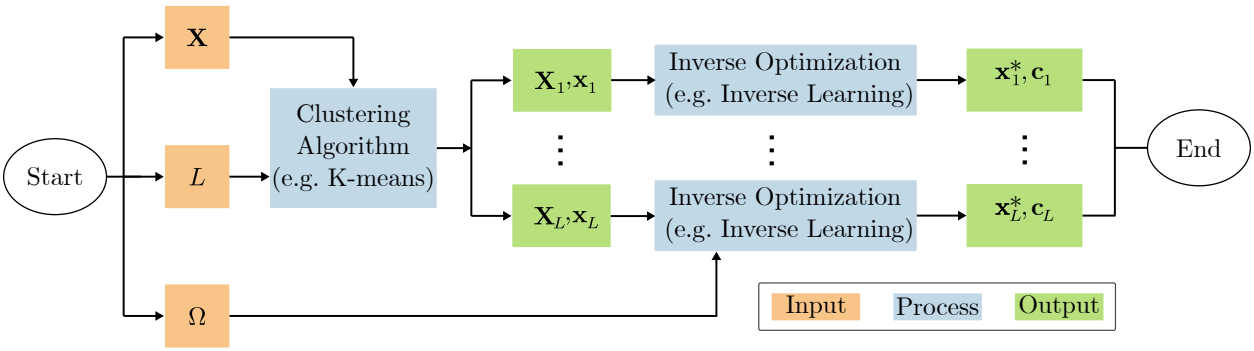


Figure 3 **Overview of the SEQ-MLIO methodology. This approach clusters the observations based on a given loss metric and finds inverse optimal solutions for each cluster.**

assumptions that rudimentary machine learning models are based on, which do not hold in general when $\Omega \neq \mathbb{R}^n$. Therefore, the SEQ-MLIO partition of $\mathbf{X}_0$ is indifferent to Ω, and additional available information on Ω does not affect the resulting partition and only affects the learned optimal solutions. To address this issue, in what follows, we consider another solution approach that captures Ω in the clustering algorithm.

4.2. An Embedded Approach to Solving MLIO

To improve on the solution found by SEQ-MLIO, one can either improve the partition or find cluster representatives that decrease the overall loss. Because IO always learns cluster representatives with the minimum loss to the observations within a cluster, finding improved partitions is a natural direction to reduce the total loss. In this section, we provide a modified heuristics method that is capable of providing partial optimality guarantees.

To start, consider the feasible set of MLIO that contains all possible partitions of $\mathbf{X}$ for a given number of clusters. Developing a heuristic that learns optimal or local optimal solutions is a daunting task. For such problems, previous literature has considered the notion of *partial optimal* solutions as defined in Definition 1 (Chakraborty and Das 2017, Selim and Ismail 1984).

DEFINITION 1. Let $\mathbf{D}(\mathbf{X}_1,\ldots,\mathbf{X}_L,\mathbf{x}_1,\ldots,\mathbf{x}_L) = \sum_{i=1}^{L} \mathcal{D}(\mathbf{x}_i,\mathbf{X}^i)$. The solution $(\mathbf{X}_1,\ldots,\mathbf{X}_L,\mathbf{x}_1,\ldots,\mathbf{x}_L,\mathbf{c}_1,\ldots,\mathbf{c}_L)$ is a partial optimal solution for MLIO if it satisfies the following:

(i) $\mathbf{D}(\mathbf{X}_1,\ldots,\mathbf{X}_L,\mathbf{x}_1,\ldots,\mathbf{x}_L) \leq \mathbf{D}(\mathbf{X}'_1,\ldots,\mathbf{X}'_L,\mathbf{x}_1,\ldots,\mathbf{x}_L)$ for any $\{\mathbf{X}'_1,\ldots,\mathbf{X}'_L\} \in \mathbf{P}_L(\mathbf{X})$.

(ii) $\mathbf{D}(\mathbf{X}_1,\ldots,\mathbf{X}_L,\mathbf{x}_1,\ldots,\mathbf{x}_L) \leq \mathbf{D}(\mathbf{X}_1,\ldots,\mathbf{X}_L,\mathbf{x}'_1,\ldots,\mathbf{x}'_L)$ for any solution set $\{\mathbf{x}'_1,\ldots,\mathbf{x}'_L\} \subseteq \Omega^{opt}$.

Definition 1 provides a measure of quality for a proposed solution to MLIO. A solution is partial optimal if the sub-problems of finding the best set of clusters with fixed cluster representatives (which corresponds to part (i) of the definition) and finding the best cluster representatives with a fixed set of clusters do not yield better solutions (which corresponds to part (ii) of the definition) in terms of the overall metric $\mathbf{D}$.

Using Definition 1, we propose an algorithm that converges to a partial optimal solution for MLIO. To this end, we embed the inverse learning model (which provides optimality guarantees for the learned parameters and solutions) into the clustering model (which handles the partitioning decisions) to return optimal solutions as cluster representatives. Such an approach will allow the model to update the partition of $\mathbf{X}_0$ in a direction that reduces the total loss. Instead of finding point estimators in $\mathbb{R}^n$ that minimize loss to the observed decisions, our proposed embedded algorithm aims to find solutions contained in Ω^{opt} and reduce the distance between found solutions and the decisions in the clusters until a local optimum is reached. We show the algorithm concludes after a finite number of iterations and discuss desirable properties regarding the final output of the algorithm. We then compare the results of these solution models in terms of optimal objective and run time with MLIO in Sections 4.3 and 5.

The embedded approach EMB-MLIO is depicted in Figure 4 similar to the SEQ-MLIO in Section 4.1. As indicated in Figure 4, additional steps for the EMB-MLIO approach can improve the quality of the final solution relative to the SEQ-MLIO approach. Details of the EMB-MLIO approach are provided in Algorithm 1. The algorithm starts with a distance-based clustering of the observations (e.g., similar to the K-means algorithm) and iteratively solves the IO problem for each cluster and updates the clusters based on the metric value of each observed decision to the newly learned optimal solutions. The algorithm terminates if no changes are made to the clusters or if the total loss does not decrease. We first outline the feasibility of the learned solutions from Algorithm 1 for MLIO and show that for any observed decision set $\mathbf{X}_0$, EMB-MLIO returns a solution with a smaller total loss ($\mathcal{D}$) relative to SEQ-MLIO.

PROPOSITION 5. *The solutions* $(\mathbf{X}_0^1, \ldots, \mathbf{X}_0^L, \mathbf{x}_1^*, \ldots, \mathbf{x}_L^*, \mathbf{c}_1, \ldots, \mathbf{c}_L)$ *from each iteration of Algorithm 1 are feasible for MLIO.*

Proposition 5 states that the solution returned using EMB-MLIO (Algorithm 1) contains optimal solutions for each cluster. In the following proposition, we go one step further by showing EMB-MLIO always provides a solution that reduces the total loss in comparison to the solution obtained from SEQ-MLIO.

PROPOSITION 6. *Let* $(\{\mathbf{X}_1^S, \ldots, \mathbf{X}_L^S\}, \{\mathbf{x}_1^S, \ldots, \mathbf{x}_L^S\}, \{\mathbf{c}_1^S, \ldots, \mathbf{c}_L^S\})$ *be the MLIO feasible solution resulting from the SEQ-MLIO approach. Then,* $\exists$ $(\{\mathbf{X}_1^E, \ldots, \mathbf{X}_L^E\}, \{\mathbf{x}_1^E, \ldots, \mathbf{x}_L^E\}, \{\mathbf{c}_1^E, \ldots, \mathbf{c}_L^E\})$ *can be achieved from the EMB-MLIO approach using Algorithm 1 such that* $\sum_{l=1}^{L} \mathcal{D}(\mathbf{x}_l^E, \mathbf{X}_l^E) \leq \sum_{l=1}^{L} \mathcal{D}(\mathbf{x}_l^S, \mathbf{X}_l^S)$.

EMB-MLIO is capable of finding local improvements, if any are available, when the solution from SEQ-MLIO is input to Algorithm 1. Having established that Algorithm 1 is well-defined

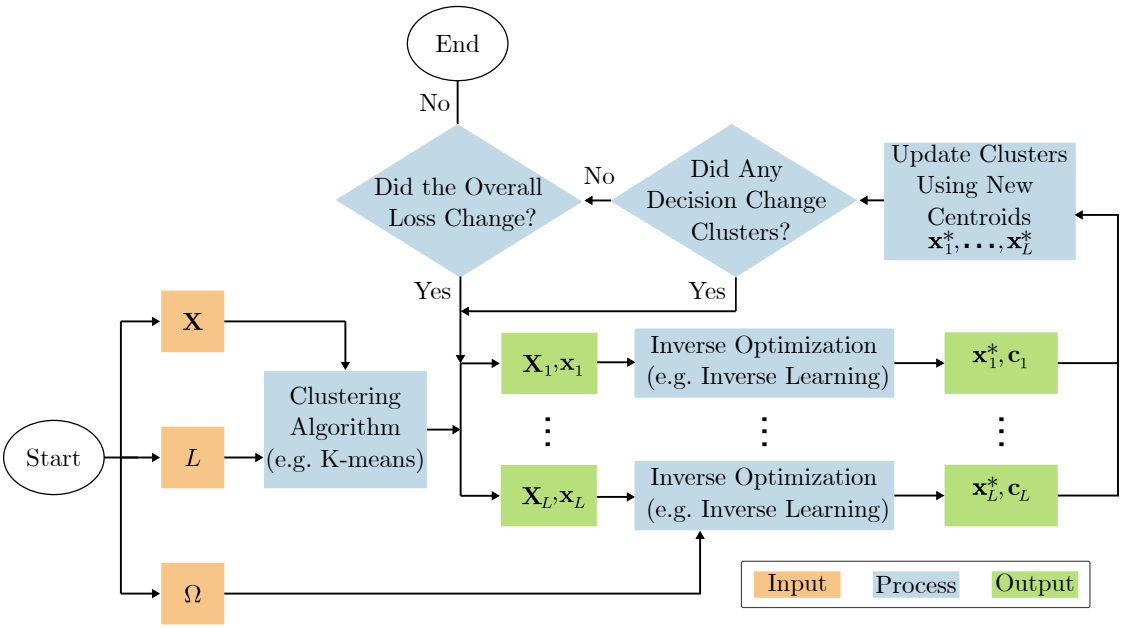


Figure 4 **Overview of the EMB-MLIO methodology. This approach clusters the observations based on some loss metric and finds inverse optimal solutions for each cluster, whereas an embedded inverse optimization model guides the clustering updates in each iteration until the termination conditions are satisfied.**

through Propositions 5 and 6, we now turn our attention to explaining the convergence and quality of the final solution of Algorithm 1. We reiterate that because Algorithm 1 is a K-means type algorithm, one can prove it terminates in a finite number of iterations. To prove this result, we first show in Lemma 1 that each possible partition of $\mathbf{X}$ is visited at most once through the course of Algorithm 1.

LEMMA 1. *Any* $\bar{\mathbf{X}} = \{\mathbf{X}_1, \ldots, \mathbf{X}_L\} \in \mathbf{P}_L(\mathbf{X})$ *is visited at most once through in Algorithm 1.*

Using the results of Lemma 1 and the fact that Algorithm 1 is strictly decreasing in terms of total loss between iterations, we show in Proposition 7 that Algorithm 1 converges to a solution in a finite number of iterations. We note the process of proving the convergence of Algorithm 1 is similar to the work of Selim and Ismail (1984), who show similar results for the K-means algorithm.

PROPOSITION 7. *Algorithm 1 converges in* $\rho \leq |\mathbf{P}_L(\mathbf{X})|$ *iterations.*

Proposition 7 shows Algorithm 1 always converges to a solution in a finite number of iterations, which is capped at the number of possible partitions corresponding to the desirable number of clusters L. Note this result per se does not guarantee the performance of the final solution of Algorithm 1. Next, in Theorem 2, we show the output of the algorithm is a partial optimal solution for MLIO.

THEOREM 2. *Let* $(\mathbf{X}^1, ..., \mathbf{X}^L, \mathbf{x}_1, ..., \mathbf{x}_L, \mathbf{c}_1, ..., \mathbf{c}_L)$ *be the output of Algorithm 1 on* $(\mathbf{X}_0, \Omega, L)$. *Then,* $(\mathbf{X}^1, ..., \mathbf{X}^L, \mathbf{x}_1, ..., \mathbf{x}_L, \mathbf{c}_1, ..., \mathbf{c}_L)$ *is a partial optimal solution for MLIO.*

Algorithm 1 EMB-MLIO for Clustering Decisions

INPUT: $\mathbf{X}$, Ω, L

OUTPUT: $\mathbf{X}_1, ..., \mathbf{X}_L$, $\mathbf{x}_1, ..., \mathbf{x}_L$, $\mathbf{c}_1, ..., \mathbf{c}_L$

Select Initial Partition $\mathbf{X}_1, ..., \mathbf{X}_L$

Step:

for $i \in \{1, ..., L\}$ **do**

  $\mathbf{c}_i, \mathbf{y}_i, \mathbf{x}_i, \mathbf{E}_i \leftarrow$Solve $\text{IO}_i(\mathbf{X}_i, \Omega)$

end for

for $\mathbf{x} \in \mathbf{X}$ **do**

  Find $i^* = \arg\min \{\mathcal{D}(\mathbf{x}, \mathbf{x}_i) \quad \forall i \in \mathcal{L}\}$

  Add $\mathbf{x}$ to $\mathbf{X}^{New}$

end for

$\bar{\mathbf{X}} \leftarrow \{\mathbf{X}_1, \ldots, \mathbf{X}_L\}$, $\bar{\mathbf{X}}^{New} \leftarrow \{\mathbf{X}_1^{New}, \ldots, \mathbf{X}_L^{New}\}$

if $\bar{\mathbf{X}}^{New} = \bar{\mathbf{X}}$ **then**

  $\mathbf{X}_1, ..., \mathbf{X}_L \leftarrow \mathbf{X}_1^{New}, ..., \mathbf{X}_L^{New}$

  STOP

end if

if $\mathbf{D}(\mathbf{X}_1^{New}, \ldots, \mathbf{X}_L^{New}, \mathbf{x}_1, \ldots, \mathbf{x}_L) = \mathbf{D}(\mathbf{X}_1, \ldots, \mathbf{X}_L, \mathbf{x}_1, \ldots, \mathbf{x}_L)$ **then**

  $\mathbf{X}_1, ..., \mathbf{X}_L \leftarrow \mathbf{X}_1^{New}, ..., \mathbf{X}_L^{New}$

  STOP

else

  $\mathbf{X}_1, ..., \mathbf{X}_L \leftarrow \mathbf{X}_1^{New}, ..., \mathbf{X}_L^{New}$

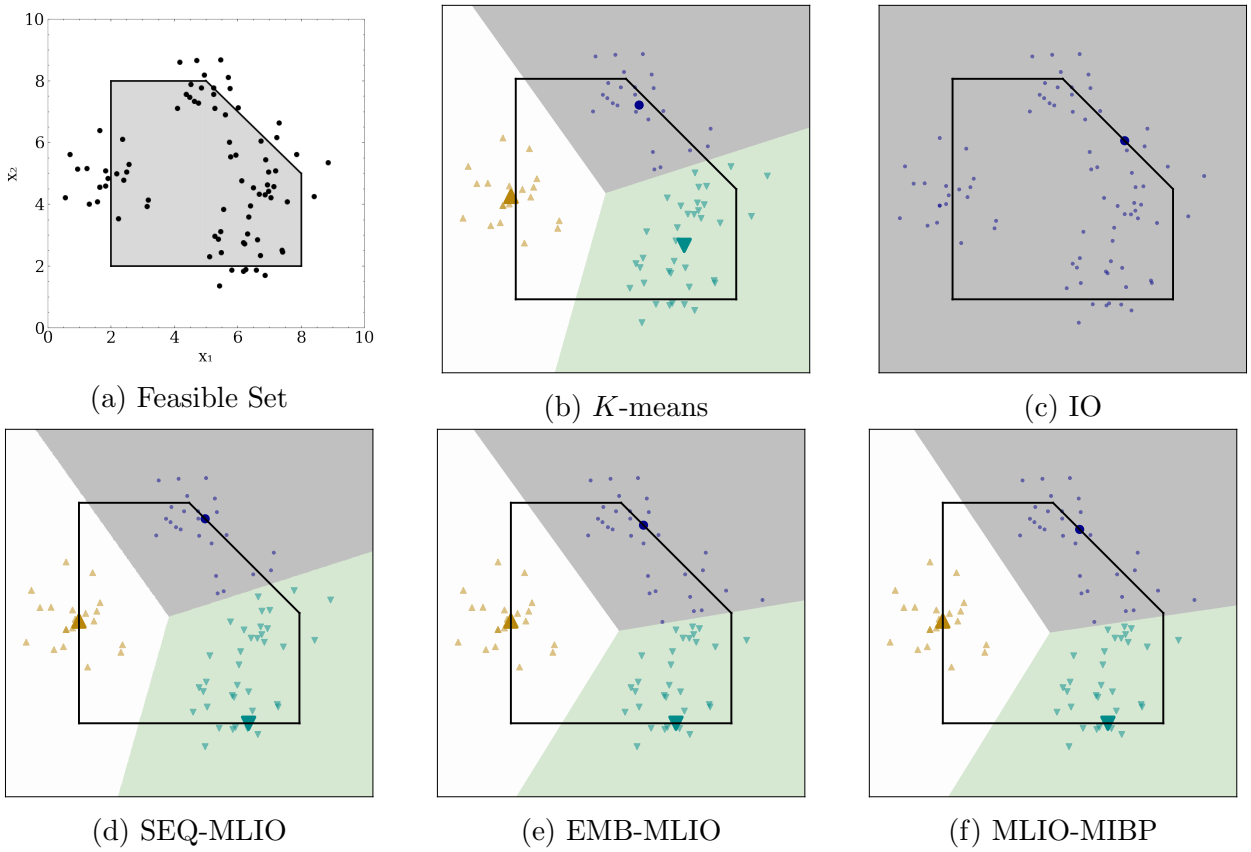
  Go to **Step**.

end if

Based on the above discussions, the resulting partition from Algorithm 1 has the desirable characteristic of being a partial optimal solution for MLIO. Additionally, using arguments similar to those in the literature (Selim and Ismail 1984), one can show the final MLIO solution found from Algorithm 1 is a local optimal solution of MLIO for a polyhedral Ω. Stated differently, EMB-MLIO is capable of efficiently providing an MLIO solution with desirable characteristics.

4.3. Numerical Illustration

We now illustrate our proposed models through a bi-dimensional numerical example. We consider a fixed feasible set Ω representing known constraints for healthy lifestyles, and assume a number of observed decisions scattered over this fixed feasible set. We then compare how different models perform in partitioning these given decisions, learning optimal solutions, and recovering optimization



(a) Feasible Set (b) K-means (c) IO

(d) SEQ-MLIO (e) EMB-MLIO (f) MLIO-MIBP

Figure 5 **Comparison of partitioning results using MLIO-MIBP, SEQ-MLIO and EMB-MLIO as well as stand-alone inverse optimization and clustering models for a two-dimensional example problem. Combining machine learning and inverse optimization approaches allows MLIO-MIBP and EMB-MLIO to modify the color mapping of clusters and representative solutions to adhere to knowledge gained using Ω.**

models for these observed decisions.

EXAMPLE 1. Let $\Omega = \{(x_1, x_2) \mid 1 \leq x_1 \leq 5, \quad 1 \leq x_2 \leq 10, \quad x_1 + x_2 \leq 15\}$ be a bi-dimensional polyhedral feasible set of a constrained decision making environment with unknown cost vectors. This fixed and known feasible set represents the constraints for healthy lifestyles that MLIO models use to provide a partitioning of the observed decisions for a given number of clusters. Additionally, a set of 80 observed decisions was randomly generated that included both feasible and infeasible points for Ω. The observed decisions and Ω are shown in Figure 5(a). For linear problems, the relationship between optimality and vicinity to the boundary of Ω will cause the MLIO models to provide an alternative partitioning in comparison to stand-alone unsupervised clustering models, as shown in Figure 5.

Figure 5 and Table 2 compare the results of employing (1) stand-alone inverse optimization, (2) K-means clustering, (3) the $\text{MLIO}_{\text{MLIO-MIBP}}$ model, (4) the benchmark model SEQ-MLIO, and (5) the heuristics solution model EMB-MLIO for the example case of clustering the given 80 decisions

to three groups using the norm 2 distance between observed decisions and the optimal solution as the metric. Figure 5(a) shows the feasible set and the observed decisions, and the other subfigures show the result from applying each model to the observed decisions and the given feasible set for finding three groups. Figure 5(b) shows the clustering result of employing the conventional K-means algorithm to $\mathbf{X}$. Note K-means is blind to Ω. On the other hand, Figure 5(c) represents the outcome of applying the inverse learning model to the observed decisions while assuming (for comparison purposes) the observed decisions are homogeneous; the inverse learning model learns an optimal solution to FO that minimizes the defined metric for the case of only one cluster, because the inverse learning model, by definition, lacks partitioning capabilities. The same effect as K-means is also seen in the benchmark model SEQ-MLIO in Figure 5(d). Although the benchmark model SEQ-MLIO captures relatively acceptable optimal solutions (in the case of this representative example), the partitioning of the observed decisions suffers from being blind to Ω. As shown in Figure 5(e), using the modified heuristics method, EMB-MLIO, results in shifting the mappings found for different clusters based on the additional knowledge provided by Ω. This finding is confirmed by the results from solving the $\text{MLIO}_{\text{MLIO-MIBP}}$ model, which partitions the observed decisions in a manner that benefits from the given constraint data and is optimal based on the definition of MLIO. This feature is particularly important because EMB-MLIO is solved in a fraction of the time required for solving $\text{MLIO}_{\text{MLIO-MIBP}}$. All these models—except $\text{MLIO}_{\text{MLIO-MIBP}}$—solve the bi-dimensional example in a fraction of a second, including the heuristics models SEQ-MLIO and EMB-MLIO with EMB-MLIO finding the optimal solution that $\text{MLIO}_{\text{MLIO-MIBP}}$ finds. As shown in the table, whereas K-means results in the least amount of total distance to the observed decisions, the centroids provided by K-means are not optimal solutions in general.

Based on the results of the representative bi-dimensional example in this section, we observe that using the heuristic model EMB-MLIO is significantly faster than and can be as capable as MLIO at finding optimal solutions for simple problems. In what follows, we apply the EMB-MLIO model as a surrogate for MLIO in the setting of the diet recommendation problem for a set of non-homogeneous dietary behavior data. We compare the results of the EMB-MLIO model with the benchmark model SEQ-MLIO and stand-alone unsupervised learning models such as K-means. We provide insights into the parallels between unsupervised clustering problems and the MLIO problem and compare the resulting recommended diets based on the known expert data.

5. Data-Driven Diet Recommendation

In this section, we apply the MLIO model and its two solution approaches to the NHANES dataset. In Section 5.1, we briefly discuss the data used as the source for the observed dietary decisions and the constraint knowledge that forms the fixed and known Ω for the partially known optimization

Table 2 Comparison of model performances for a representative bi-dimensional example in the case of three clusters. EMB-MLIO finds the same solution that MLIO-MIBP finds at optimality. EMB-MLIO is solved within comparable times to the other solution models. K-means is not built for recovering the cost vectors, and both K-means and IO provide inferior optimality gap values based on the optimal solution found by MLIO-MIBP. Although the gap is slightly higher for SEQ-MLIO, EMB-MLIO reduces the gap to zero by finding the same optimal solution as MLIO-MIBP in roughly 0.5 seconds.

Model	Learned Centroids	Recovered Cost Vectors	Total Optimality Gap	Total Distance to Observed Decisions	Solution Time (Sec.)
K-means	$(1.9, 4.8)$ $(5.4, 7.3)$ $(6.6, 3.5)$	NA	1.75	96.82	0.025
IO	$(6.7, 7.3)$	$(0.5, 0.5)$	10.86	315.1	0.121
SEQ-MLIO	$(2, 4.8)$ $(5.4, 7.6)$ $(6.6, 2)$	$(-1.0, 0.0)$ $(0.5, 0.5)$ $(0.0, -1.0)$	0.32	136.7	0.188
EMB-MLIO	$(2, 4.8)$ $(5.5, 7.4)$ $(6.5, 2)$	$(-1.0, 0.0)$ $(0.5, 0.5)$ $(0.0, -1.0)$	0.00	135.2	0.594
MLIO-MIBP (optimal)	$(2, 4.8)$ $(5.5, 7.4)$ $(6.5, 2)$	$(-1.0, 0.0)$ $(0.5, 0.5)$ $(0.0, -1.0)$	0.00	135.2	>100

models. We then train models using EMB-MLIO, SEQ-MLIO, and K-means methods to partition the observed decisions. In Section 5.2, we provide comparisons on how these two models perform in partitioning the observed decisions to different groups and discuss relevant results.

The data-driven diet recommendation problem we consider contains different nutritional constraints that form the known and fixed Ω showcasing nutritional constraints that are extensively used in the literature (see, e.g., Garille and Gass 2001). We base this problem on the nutritional constraints of a popular diet for reducing hypertension in adults, where the dietary preferences of the patients are unknown (cost vectors). We use a set of daily food intake data from patients to recover their dietary preferences and recover optimization problems such that each optimization problem is tailored to a group of similar patients based on their dietary behaviors. To do so, we employ the EMB-MLIO model to partition the observed behaviors into a given number of clusters, recover the dietary preferences of the patients within each cluster, and recommend dietary choices for each group of similar patients. These recommendations will then have the desirable quality of minimizing in-group distances within each cluster and adhering optimally to the constraints of the DASH diet. For comparison, we consider the K-means clustering model and the SEQ-MLIO heuristics model as baseline benchmark models and base our performance metrics on the in-group

distances of the observed decisions and behaviors of the patients in the same group with the recommended diets provided by the EMB-MLIO and SEQ-MLIO models in the form of optimal solutions to optimization problems and their nutritional qualities.

5.1. Data

We draw our analysis from several datasets. To incorporate the shared feasible set Ω that represents the known constraints in the diet recommendation problem, we use the recommendations of the Dietary Approaches to Stop Hypertension (DASH) eating plan (DASH 2018) and form the constraints based on lower and upper bound nutritional bounds. These bounds are primarily based on evaluating the lower- and upper-bounds provided by the DASH eating plan for different food groups. Additionally, the forward optimization problem is capable of containing a myriad of other types of constraints (e.g., food-group-serving constraints), and the application of MLIO models is not contingent on the presence of specific types of constraints in Ω. The observations for the application of the diet recommendation problem are individuals' daily food intakes, gathered from the open-access data from the National Health and Nutrition Examination Survey (NHANES) dataset (CDC 2020). The reader is referred to the corresponding GitHub repository (Ahmadi et al. 2020b) for further details about this dataset.

Consider, from a more technical standpoint, the structure of FO in formulation (1), the variables of FO include the amount of daily intake for each food item, and the cost vector representing the preferences of the patients in each cluster. We allow non-homogeneity in observed decisions in terms of their dietary behavior and use the known constraints and the observed behaviors to cluster the decisions to homogeneous groups. The left-hand-side constraint matrix ($\mathbf{A}$) is the amount of nutrients in a serving of each of the food items for each nutrient, and each nutrient has a lower- and upper-bound constraint. The right-hand-side constraint matrix ($\mathbf{b}$) contains the bounds for each nutrient. These bounds are interpreted from the recommendation of the DASH eating plan (DASH 2018).

For the analysis that follows, we consider a set of 900 patients with similar demographics and perform an 80-20 split of the data for training and testing purposes. Then, EMB-MLIO, SEQ-MLIO, and K-means models are trained with 720 data points. We vary the number of clusters between 1 and 20 to observe the behavior of all models. We note the number of clusters indicates the number of different optimal solutions and cost vectors that the EMB-MLIO and SEQ-MLIO models recover for the problem. We spotlight insights into how the EMB-MLIO models approach the partitioning problem differently in comparison to K-means and discuss various features of the models.

5.2. Results and Discussions

In this section, we provide the results of training the EMB-MLIO and the benchmark model SEQ-MLIO on the observed dietary behaviors of the patients described in the previous section. To compare these results against each other, we consider the results of the models for different food items (as the decision variables of the optimization problem) and the nutrient values (as the main constituents of Ω) and compare them with the results of stand-alone applications of machine learning (K-means) clustering model (K-means). (Note that for the purposes of the analyses in this section, we use EMB-MLIO instead of $\text{MLIO}_{\text{MLIO-MIBP}}$ for the sake of computational efficiency.)

Figure 6 compares the total in-group distances between observed decisions for the EMB-MLIO, SEQ-MLIO, and K-means models. For different values of the numbers of clusters in the partition, EMB-MLIO outperforms SEQ-MLIO for the training data. In comparison to K-means, whereas MLIO models perform on a subset of $\mathbb{R}^n$ and provide solutions that are farther from the observations than K-means, they provide optimal solutions over the known Ω. Figure 6 shows the increase in the loss in comparison to K-means is not large. Additionally, the optimality-gap comparison figure shows how K-means does not provide optimal solutions with regard to Ω when proposing solutions for each cluster. We also note that, similar to conventional unsupervised learning models, this figure can also be used to get a notion of the appropriate number of clusters. In this case, because the curves are decreasing, one can either consider four clusters as an elbow point or consider 20 clusters for analysis. In what follows, we provide results for 20 clusters to facilitate comparisons across models.

An important aspect of the diet recommendation problem is the nutritional quality of the diets generated by the algorithm. Thus, we compare the results of the MLIO model with K-means models. As expected, because K-means models are naturally blind to the nutritional bounds, K-means models will only replicate the original behaviors of the patients, whereas MLIO will strike a balance between replicating behaviors and adhering to the hard constraints set by the DASH diet requirements. Figure 7 compares the nutritional values of the recommended diet for both models for 20 clusters. For better representation, results are also shown via box plots representing the 10, 25, 50, 75, and 90 quantiles. Most notably, one can observe that MLIO restricts target nutrients of the DASH diet, such as sodium and cholesterol, but allows for comparable values with the original behaviors for other nutrients. The results show using MLIO allows dietitians to easily find diets that strike a balance between patient preferences and dietary goals.

In addition to comparing the in-group distances to the centroids and analyzing the quality of the recommended diets based on their nutritional values, we can analyze the centroids learned from the applications of the EMB-MLIO model and a stand-alone unsupervised learning model such as K-means. Figure 8 compares the values of each food item for EMB-MLIO (designated as MLIO

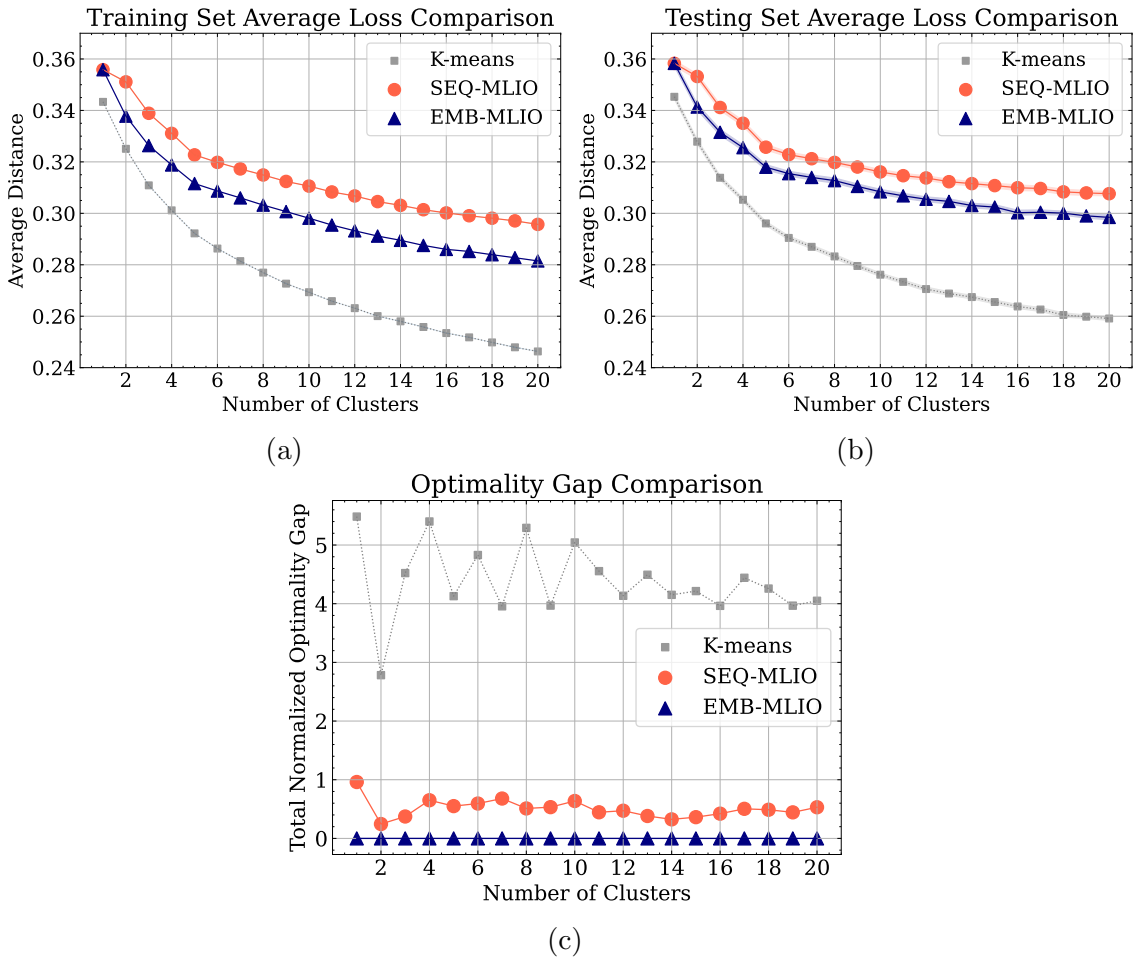


Figure 6 **The average in-group distances are calculated for different numbers of clusters using the K-means (gray), SEQ-MLIO (orange), and EMB-MLIO (navy). Figure 6a illustrates the average distance from the obtained solutions to the observed decisions within their clusters in the training data. Figure 6b represents similar values with regard to the testing data. As shown, in both cases, EMB-MLIO consistently achieves smaller distances than SEQ-MLIO. Figure 6c shows the optimality gap of these models, where EMB-MLIO achieves optimality and SEQ-MLIO averages around 0.5 in the total normalized optimality gap to the number of data points in the training set. In most cases, K-means represents a fluctuating pattern with normalized values above 4. For this analysis, training and testing samples include 720 and 180 data points of daily food intake, respectively.**

in the figure) and K-means (designated as K-means in the figure). This comparison indicates the distinctions between the K-means and MLIO approaches in their learning solutions. Most notable among the food items, one can point out the fruit and seed food groups, which MLIO recommends significantly more than K-means. This difference in increased value represents adherence to nutritional bounds in recommending nutritious food groups. Additionally, as observed from Figure 9, recommendations for fruits are quite diverse for different clusters of patients. This can be attributed to the high bounds on dietary fiber. Because different clusters exhibit different levels of

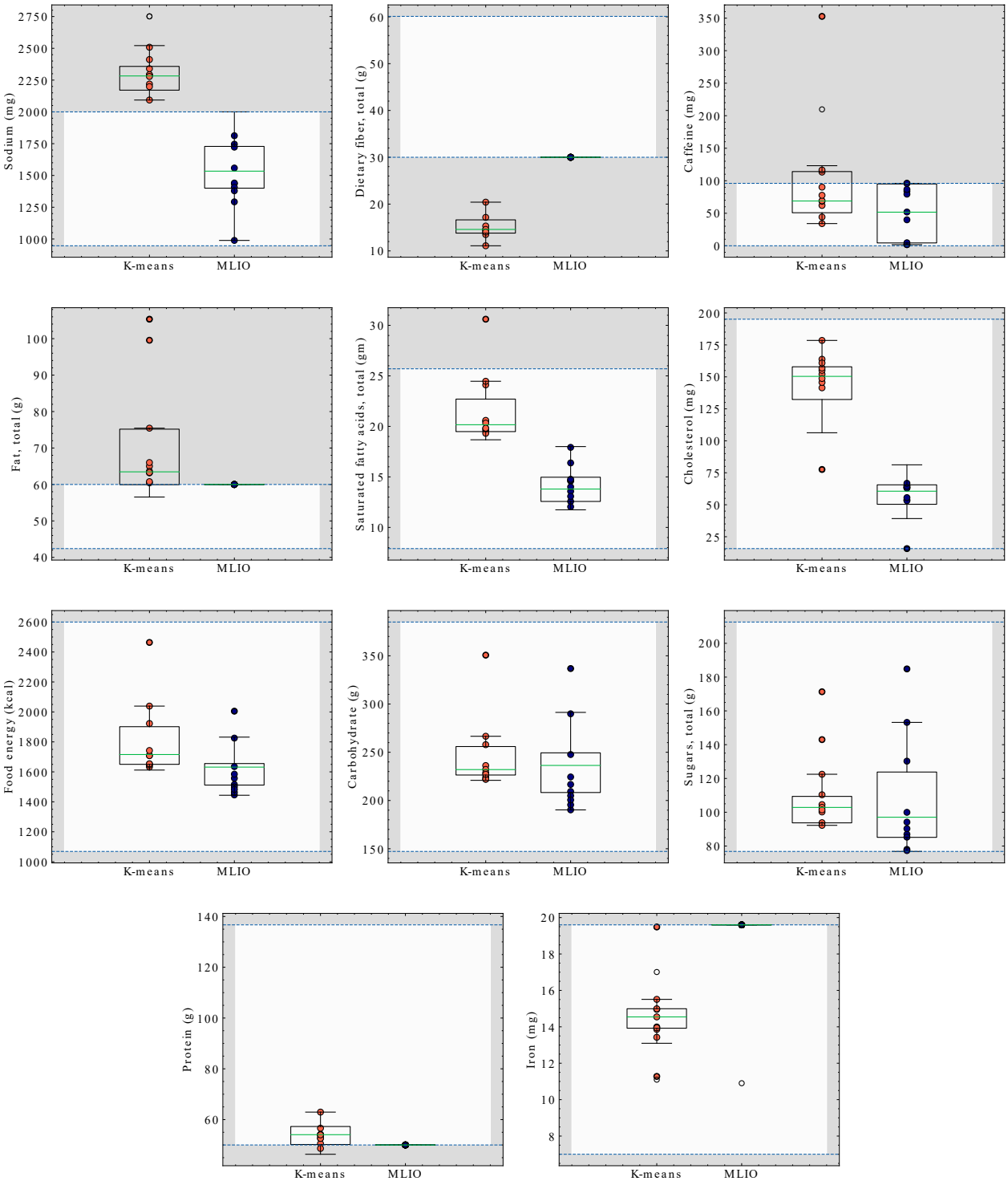


Figure 7 **Box plots for the nutritional values of the diets recommended using the K-means and the EMB-MLIO models. Nutrients with infeasible values are shaded. K-means recommends out of limit values for sodium and dietary fiber across the board, whereas EMB-MLIO recommends diets that adhere to DASH limits (the non-shaded areas in the figures). K-means recommendations do not completely adhere to DASH limits for all clusters in the cases of total fat, protein, and cholesterol.**

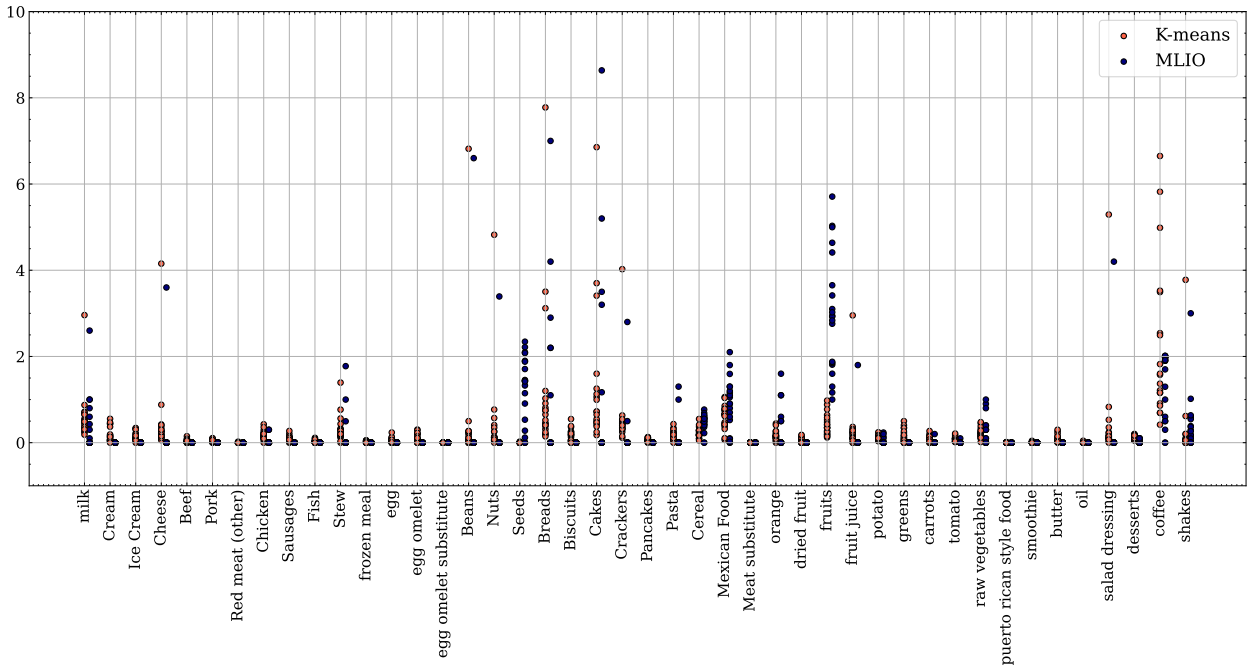


Figure 8 **Comparison of food group recommendations between the EMB-MLIO and K-means models.**

fiber, fruits are used primarily to adjust diets to the required levels imposed by the DASH diet. On the other hand, items such as cheeses, creams, and dressings are among the foods that MLIO recommends noticeably less than K-means, which also represents restricting unhealthy groups, whereas bread, crackers, and milk are among the foods that MLIO and K-means recommend to comparable extents, demonstrating MLIO's adherence to original behaviors as much as possible. The results of this figure confirm how MLIO balances between replicating original behaviors and incorporating constraint information.

The results provided in this section show the MLIO approach is capable of both (1) providing improved partitioning of observed decisions when information on healthy nutritional boundaries is available and also recovering optimization problems and (2) learning the best possible optimal solutions to such optimization problems based on the decisions partitioned in the same cluster. In the case of the diet recommendation, the recommendation system is capable of balancing between the healthy dietary constraints forming the dietary nutritional bounds and the observed behavior of the patients, and it provides diets that conform to the known dietary constraints. Additionally, we showed that, whereas MLIO restricts the unhealthy behaviors of patients, it is also capable of replicating their original preferences to the extent that the constraint-set information allows.

6. Concluding Remarks

In this paper, we present a novel, data-driven approach to generating diet recommendations that incorporate both diverse dietary preferences and healthy dietary standards. This approach unifies

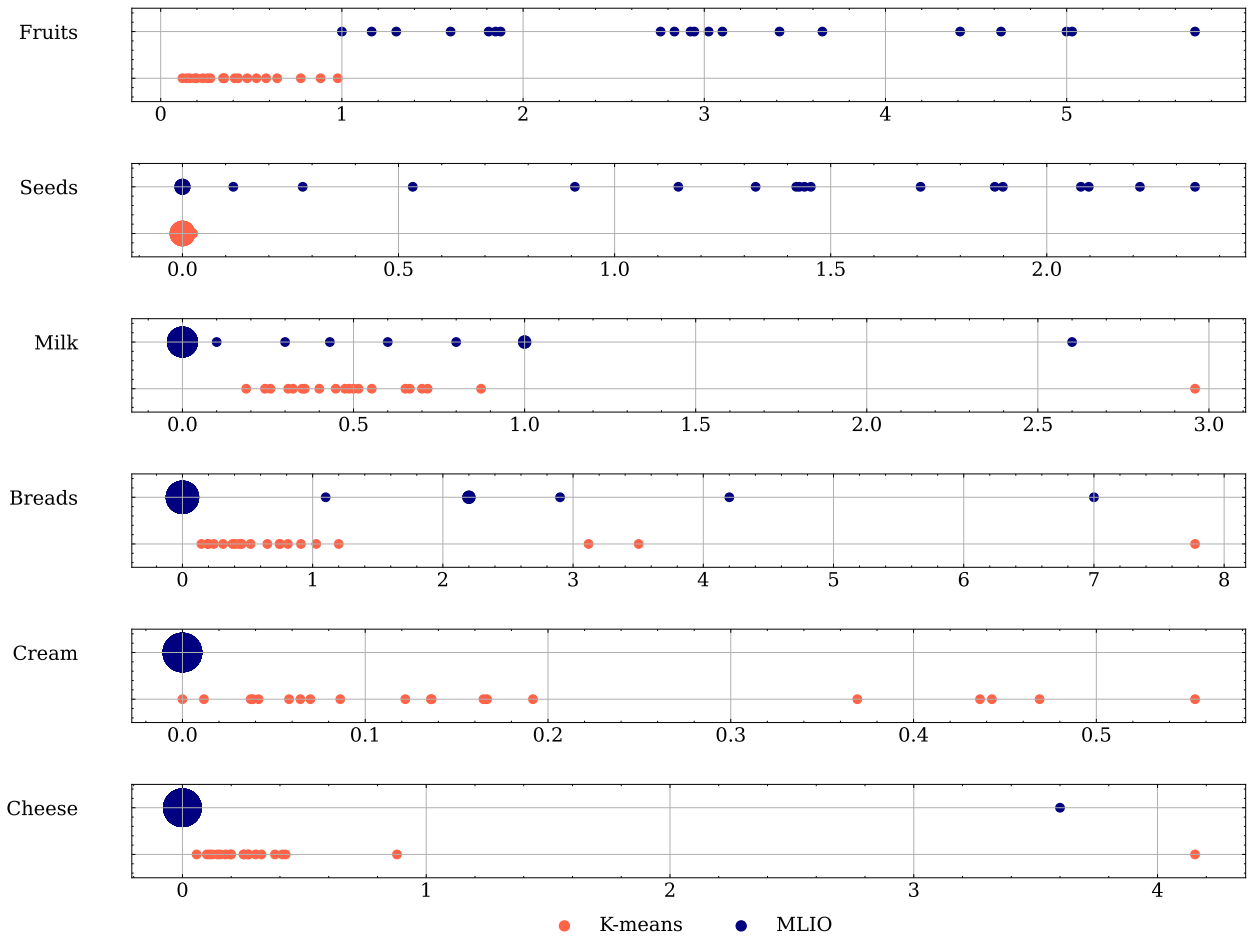


Figure 9 **Individual food-group recommendations between the EMB-MLIO and K-means models. EMB-MLIO promotes healthy food groups such as fruits and seeds, whereas K-means replicates general unhealthy behaviors such as excessive consumption of creams in certain groups. Dots with increased radii indicate a higher number of groups that have that level of recommended intake for the food group.**

inverse optimization and clustering. Under this hybrid approach, for the first time, inverse learning is embedded into a clustering scheme to recover unknown parameters and generate optimal solutions while ensuring optimality for generated solutions that act as cluster representatives. We demonstrated that sequentially applying unsupervised learning models (for partitioning of individuals) and inverse optimization models (for recovering the optimization problems) results in structurally distinct and suboptimal partitioning of observed decisions compared with solving the actual optimization problem; we proposed a solution method that provides some level of assurance over the partitioning quality. To this end, we developed a novel solution strategy that *embeds* clustering techniques and inverse optimization into each other. This strategy generalizes the clustering problem to the case where cluster representatives are subject to hard constraints and connects the generalized problem to existing inverse optimization models that recover the original optimization

problems using observed data.

The advantages of our hybrid approach are twofold. Results from applying the MLIO models to a bi-dimensional example with non-homogeneous observations over a fixed feasible set show how MLIO adapts to the known hard constraints of the problem setting. Equally important, applying MLIO models to the diet recommendation problem enables diet plan makers to incorporate group preferences by recovering missing objective functions and proposing optimal diets for different groups of individuals. Using MLIO—as opposed to conventional machine learning techniques—allows dietary constraints to play a role in sharing how observations are clustered. By using MLIO, we are able to redirect the recommendations of the systems toward diets that adhere to the nutritional limits set forth by experts, in our case, the DASH eating plans. Specifically for the diet recommendation problem, a naïve clustering model results in the replication of unhealthy behaviors of patients. By contrast, we show MLIO models are capable of recommending diets that comply with DASH nutritional standards.

The MLIO approach described in this paper represents a step forward in modeling clustering problems with the goal of recommending optimal decisions for groups. This new approach will allow for data-driven decision support in applications such as the diet recommendation problem discussed in Section 5. The MLIO approach can be used as the basis for developing an interactive recommendation tool that provides dietary recommendations based on their dietary history and dietary constraints catered to their needs.

Acknowledgments

We gratefully acknowledge the financial support from the Johns Hopkins Discovery Award (2021–2023) and the Johns Hopkins Malone Center for Engineering in Healthcare Seed Grant (2020–2022). We appreciate the comments and suggestions from Fardin Ganjkhanloo, Julien Grand-Clément, Huseyin Gurkan, Todd McNutt, and Nasrin Yousefi. We have also benefited from the feedback from seminar participants at the Johns Hopkins University's Center for Systems Science and Engineering and the Department of Applied Mathematics and Statistics, and session participants at 2020 INFORMS Annual Meeting, 2021 ACM Conference on Health, Inference, and Learning (CHIL), 2021 MSOM Annual Conference, and 2022 CORS/INFORMS International Conference.

Appendix: Proofs

Proof of Proposition 1 Let $\mathbf{X}_1,\dots,\mathbf{X}_L$ be some partition of $\mathbf{X}_0$. $\text{IO}_1(\Omega,\mathbf{X}_1)$, …, $\text{IO}_L(\Omega,\mathbf{X}_L)$ are all feasible and have at least one optimal solution set. Let $(\mathbf{c}_1,\mathbf{y}_1,\mathbf{x}_1^*,\mathbf{E}_1),\dots,(\mathbf{c}_L,\mathbf{y}_L,\mathbf{x}_L^*,\mathbf{E}_L)$ be optimal solutions for $\text{IO}_1(\Omega,\mathbf{X}_1)$, …,$\text{IO}_L(\Omega,\mathbf{X}_L)$, respectively. Because $\mathbf{x}_l^*$ are optimal for $\text{FO}(\mathbf{c}_l,\Omega)$ $\forall l\in\mathcal{L}$, the solution $(\mathbf{x}_1^*,\dots,\mathbf{x}_L^*,\mathbf{c}_1,\dots,\mathbf{c}_L)$ satisfies constraint (3b). Then, $\mathbf{X}_1,\dots,\mathbf{X}_L,\mathbf{x}_1^*,\dots,\mathbf{x}_L^*,\mathbf{c}_1,\dots,\mathbf{c}_L$ is a feasible solution for $\text{MLIO}(\mathbf{X}_0,\Omega,L)$. *Q.E.D.*

Proof of Proposition 2 Let $\mathbf{X}_1,\dots,\mathbf{X}_L\in\mathbf{P}_L(\mathbf{X})$ be a partition of $\mathbf{X}_0$ to L clusters. Due to feasibility of IO, $\mathbf{x}_1,\dots,\mathbf{x}_L$ exist such that solution sets $(\{\mathbf{c}_1,\mathbf{y}_1,\mathbf{x}_1,\mathbf{E}_1\},\dots,\{\mathbf{c}_L,\mathbf{y}_L,\mathbf{x}_L,\mathbf{E}_L\}$ are optimal for $\text{IO}(\mathbf{X}_1,\Omega),\dots,\text{IO}(\mathbf{X}_L,\Omega)$ respectively. As such, $\mathbf{X}_1,\dots,\mathbf{X}_L,\mathbf{x}_1,\dots,\mathbf{x}_L,\mathbf{c}_1,\dots,\mathbf{c}_L$ is feasible for $\text{MLIO}(\mathbf{X}_0,\Omega,L)$. Because the same is true for any partition in $\mathbf{P}_L(\mathbf{X})$, $\mathbf{P}_L(\mathbf{X})\subseteq\hat{\Psi}_L$. *Q.E.D.*

Proof of Proposition 3 Let $(\mathbf{x}^*,\mathbf{v}^*,\mathbf{Y}^*,\mathbf{C}^*,\mathbf{E}^*)$ be optimal for $\text{MLIO-MIBP}(\mathbf{X}_0,\Omega,L)$. We first show that for each $l\in\mathcal{L}$, $(\mathbf{C}_l^*,\mathbf{Y}_l^*,\mathbf{x}_l^*,\hat{\mathbf{E}}_l^*)$ is feasible for $\text{IO}(\mathbf{X}_l^*,\Omega)$, where $(\mathbf{C}_l^*,\mathbf{Y}_l^*,\mathbf{x}_l^*,\hat{\mathbf{E}}_l^*)$ are defined similar to the statement of the proposition. This can be verified by noting constraints of MLIO-MIBP are equivalent to IO for each value of l. Now, assume to the contrary that for some $l\in\mathcal{L}$, $(\mathbf{C}_l^*,\mathbf{Y}_l^*,\mathbf{x}_l^*,\hat{\mathbf{E}}_l^*)$ is not optimal for $\text{IO}(\mathbf{X}_l^*,\Omega)$. Then, some solution $(\mathbf{C}_l',\mathbf{Y}_l',\mathbf{x}_l',\mathbf{E}_l')$ exist that is optimal for $\text{IO}(\mathbf{X}_l^*,\Omega)$, such that $\mathcal{D}(\mathbf{X}_l^*,\mathbf{x}_l')<\mathcal{D}(\mathbf{X}_l^*,\mathbf{x}_l^*)$. However, we can build another solution to $\text{MLIO-MIBP}(\mathbf{X}_0,\Omega,L)$ by replacing $(\mathbf{C}_l^*,\mathbf{Y}_l^*,\mathbf{x}_l^*,\hat{\mathbf{E}}_l^*)$ with $(\mathbf{C}_l',\mathbf{Y}_l',\mathbf{x}_l',\mathbf{E}_l')$ in the solution $(\mathbf{x}^*,\mathbf{v}^*,\mathbf{Y}^*,\mathbf{C}^*,\mathbf{E}^*)$. Because this new solution is feasible for $\text{MLIO-MIBP}(\mathbf{X}_0,\Omega,L)$, this is a contradiction to $(\mathbf{x}^*,\mathbf{v}^*,\mathbf{Y}^*,\mathbf{C}^*,\mathbf{E}^*)$ being optimal for $\text{MLIO-MIBP}(\mathbf{X}_0,\Omega,L)$ as we have found a new solution with a smaller objective value. *Q.E.D.*

Proof of Theorem 1 Let $(\mathbf{x}^*,\mathbf{v}^*,\mathbf{Y}^*,\mathbf{C}^*,\mathbf{E}^*)$ be optimal for $\text{MLIO-MIBP}(\mathbf{X}_0,\Omega,L)$. Additionally, for $l\in\mathcal{L}$, let $\mathbf{X}_l$ be the set of all elements of $\mathbf{X}_0$ such that $\mathbf{x}^k\in\mathbf{X}_l$ if and only if $v_{k,l}^*=1$. Considering that the solution $(\mathbf{X}^*,\mathbf{v}^*,\mathbf{Y}^*,\mathbf{C}^*,\mathbf{E}^*)$ is feasible for $\text{MLIO-MIBP}(\mathbf{X}_0,\Omega,L)$; it satisfies (4f). Therefore, we have $\bigcup_{l=1}^{L}\mathbf{X}_l=\mathbf{X}_0$, and as such, $\{\mathbf{X}_1,\dots,\mathbf{X}_L\}$ is a partition of $\mathbf{X}_0$. Additionally, constraints (4a), (4b), and (4c) ensure for all $l\in\mathcal{L}$, $\mathbf{x}_l^*$ is optimal for $\text{FO}(\mathbf{C}_l^*,\Omega)$. Therefore, $\forall l\in\mathcal{L}$, we have $\mathbf{x}_l^*\in\Omega^{opt}(\mathbf{c}_l)$ and as such, $(\{\mathbf{X}_1,\dots,\mathbf{X}_L\},\mathbf{x}^*,\mathbf{C}^*)$ is feasible for MLIO. Now, assume to the contrary that $(\{\mathbf{X}_1,\dots,\mathbf{X}_L\},\mathbf{x}^*,\mathbf{C}^*)$ is not optimal for MLIO. Then, because any solution of MLIO can be mapped to at least one solution of MLIO-MIBP by similar arguments as above, we would find a better solution for MLIO-MIBP than $(\mathbf{x}^*,\mathbf{v}^*,\mathbf{Y}^*,\mathbf{C}^*,\mathbf{E}^*)$, which is a contradiction to $(\mathbf{x}^*,\mathbf{v}^*,\mathbf{Y}^*,\mathbf{C}^*,\mathbf{E}^*)$ being optimal for $\text{MLIO-MIBP}(\mathbf{X}_0,\Omega,L)$. *Q.E.D.*

Proof of Proposition 4 Let $(\{\mathbf{X}_1,\dots,\mathbf{X}_L\},\{\mathbf{x}_1^*,\dots,\mathbf{x}_L^*\},\{\mathbf{c}_1,\dots,\mathbf{c}_L\})$ be the solution obtained from SEQ-MLIO. To show $(\{\mathbf{X}_1,\dots,\mathbf{X}_L\},\{\mathbf{x}_1^*,\dots,\mathbf{x}_L^*\},\{\mathbf{c}_1,\dots,\mathbf{c}_L\})$ is feasible for MLIO, we need to prove $\forall l\in\mathcal{L}$, $\mathbf{x}_l^*\in\Omega^{opt}(\mathbf{c}_l)$. However, notice that based on the definition of SEQ-MLIO, $\forall l\in\mathcal{L}$, $\mathbf{x}_l^*$ is part of an optimal solution to $\text{IO}(\mathbf{X}_l,\Omega)$, which is equivalent to $\mathbf{x}_l^*$ being contained in $\Omega^{opt}(\mathbf{c}_l)$. As such, the solution obtained by SEQ-MLIO is feasible for MLIO. *Q.E.D.*

Proof of Proposition 5 We first note that by definition of Algorithm 1, for the solution set $(\{\mathbf{X}_1^i,\dots,\mathbf{X}_L^i\},\{\mathbf{x}_1^i,\dots,\mathbf{x}_L^i\},\{\mathbf{c}_1^i,\dots,\mathbf{c}_L^i\})$ found in the i^{th} iteration of Algorithm 1, solutions $(\mathbf{c}_1^i,\mathbf{y}_1^i,\mathbf{x}_1^i,\mathbf{E}_1^i),\dots,(\mathbf{c}_L^i,\mathbf{y}_L^i,\mathbf{x}_L^i,\mathbf{E}_L^i)$ exist that are optimal for $\text{IO}(\Omega,\mathbf{X}_1^i)$, ..., $\text{IO}(\Omega,\mathbf{X}_L^i)$, respectively. As such,

recalling the definition of $\Omega^{opt}(\mathbf{c})$, we have $\mathbf{x}_1^i \in \Omega^{opt}(\mathbf{c}_1^i), \ldots, \mathbf{x}_L^i \in \Omega^{opt}(\mathbf{c}_L^i)$. Because $\{\mathbf{X}_1^i, \ldots, \mathbf{X}_L^i\}$ is a partition of $\mathbf{X}_0$, $(\{\mathbf{X}_1^i, \ldots, \mathbf{X}_L^i\}, \{\mathbf{x}_1^i, \ldots, \mathbf{x}_L^i\}, \{\mathbf{c}_1^i, \ldots, \mathbf{c}_L^i\})$ is feasible for MLIO. *Q.E.D.*

Proof of Proposition 6 Let $(\{\mathbf{X}_1^S, \ldots, \mathbf{X}_L^S\}, \{\mathbf{x}_1^S, \ldots, \mathbf{x}_L^S\}, \{\mathbf{c}_1^S, \ldots, \mathbf{c}_L^S\}, \{\mathbf{c}_1^S, \ldots, \mathbf{c}_L^S\})$ be the MLIO solution resulting from the SEQ-MLIO approach. One can initialize Algorithm 1 using $\{\mathbf{X}_1^S, \ldots, \mathbf{X}_L^S\}$ as the initial partition of $\mathbf{X}_0$. Considering that Algorithm 1 is strictly decreasing between iterations in terms of the total loss $\sum_{l=1}^L \mathcal{D}(\mathbf{x}_l, \mathbf{X}_l)$, if we denote the output partition and solution set of Algorithm 1 as $(\{\mathbf{X}_1^E, \ldots, \mathbf{X}_L^E\}, \{\mathbf{x}_1^E, \ldots, \mathbf{x}_L^E\}, \{\mathbf{c}_1^E, \ldots, \mathbf{c}_L^E\})$, we have $\sum_{l=1}^L \mathcal{D}(\mathbf{x}_l^E, \mathbf{X}_l^E) \leq \sum_{l=1}^L \mathcal{D}(\mathbf{x}_l^S, \mathbf{X}_l^S)$. *Q.E.D.*

Proof of Lemma 1 Assume to the contrary that $\exists\ \bar{\mathbf{X}} = \{\mathbf{X}_1, \ldots, \mathbf{X}_L\} \in \mathbf{P}_L(\mathbf{X})$ that is visited twice in Algorithm 1 on iterations i and j. For iterations i and j, we have $\{\mathbf{X}_1^i, \ldots, \mathbf{X}_L^i\} = \{\mathbf{X}_1^j, \ldots, \mathbf{X}_L^j\}$. However, this means that for iterations i and j, $\forall\ l \in \mathcal{L}$, $\exists\ (\mathbf{c}_l^i, \mathbf{y}_l^i, \mathbf{x}_l^i, \mathbf{E}_l^i)$ optimal for $\text{IO}(\Omega, \mathbf{X}_l^i)$ and $(\mathbf{c}_l^j, \mathbf{y}_l^j, \mathbf{x}_l^j, \mathbf{E}_l^j)$ optimal for $\text{IO}(\Omega, \mathbf{X}_l^j)$ where $\{\mathbf{x}_1^i, \ldots, \mathbf{x}_L^i\} = \{\mathbf{x}_1^j, \ldots, \mathbf{x}_L^j\}$. Therefore, for these iterations, we have $\sum_{l=1}^L \mathcal{D}(\mathbf{x}_l^i, \mathbf{X}_l^i) = \sum_{l=1}^L \mathcal{D}(\mathbf{x}_l^j, \mathbf{X}_l^j)$. But this is in contradiction to Algorithm 1 being strictly decreasing in terms of $\mathcal{D}$ between iterations. As such, each partition is visited at most once throughout Algorithm 1. *Q.E.D.*

Proof of Proposition 7 Using the results of Lemma 1, each potential partition $\{\mathbf{X}_1, \ldots, \mathbf{X}_L\} \in \mathbf{P}_L(\mathbf{X})$ is visited at most once in Algorithm 1. Noting the total number of possible partitions of $\mathbf{X}_0$ to L clusters is equal to $|\mathbf{P}_L(\mathbf{X})|$, the number of iterations required for Algorithm 1 to stop and return a solution is at most $|\mathbf{P}_L(\mathbf{X})|$. *Q.E.D.*

Proof of Theorem 2 Let $(\{\mathbf{X}_1, \ldots, \mathbf{X}_L\}, \{\mathbf{x}_1, \ldots, \mathbf{x}_L\}, \{\mathbf{c}_1, \ldots, \mathbf{c}_L\})$ be the final solution obtained by running Algorithm 1 on $\mathbf{X}_0$. We first note that based on the operations done on lines 5 to 11 of Algorithm 1, for each solution of EMB-MLIO in each iteration, the first condition of Definition 1 is satisfied as these lines find the best possible partition for a given set of optimal solutions in Ω^{opt}. To investigate the second condition of Definition 1, we note that for $(\{\mathbf{X}_1, \ldots, \mathbf{X}_L\}, \{\mathbf{x}_1, \ldots, \mathbf{x}_L\}, \{\mathbf{c}_1, \ldots, \mathbf{c}_L\})$, three possible cases exist:

Case 1. Stop criterion at line 15 of Algorithm 1: In this case, solving IO for the last iteration solution did not yield a better optimal solution set. As such, the original optimal solution set $\{\mathbf{x}_1, \ldots, \mathbf{x}_L\}$ was already the best possible solution set on Ω^{opt}, which is equivalent to the second condition in Definition 1.

Case 2. Stop criterion at line 19 of Algorithm 1: In this case, although updating the partition did change $\mathbf{X}_1, ..., \mathbf{X}_L$ to a different partition $\mathbf{X}_1^{New}, ..., \mathbf{X}_L^{New}$, this has not changed the total loss and we have $\mathbf{D}(\mathbf{X}_1^{New}, \ldots, \mathbf{X}_L^{New}, \mathbf{x}_1, \ldots, \mathbf{x}_L) = \mathbf{D}(\mathbf{X}_1, \ldots, \mathbf{X}_L, \mathbf{x}_1, \ldots, \mathbf{x}_L)$. Therefore, although some other partition may generate the same total loss $\mathbf{D}$ for the optimal solution set $\{\mathbf{x}_1, \ldots, \mathbf{x}_L\}$, no partition exists that provides a smaller total loss $\mathbf{D}$. This is again equivalent to the second criterion of Definition 1.

Case 3. Neither Stop criteria in Algorithm 1: In this case, due to Algorithm 1 being strictly decreasing, the final solution found from Algorithm 1 is the global optimal solution because Algorithm 1 has exhausted all possible partitions in $\mathbf{P}_L(\mathbf{X})$. Because the global optimal solution is a partial optimal solution as well, the statement holds. *Q.E.D.*

References

Ahmadi F, Ganjkhanloo F, Ghobadi K (2020a) Inverse learning: A data-driven framework to infer optimizations models. Working paper.

Ahmadi F, Ganjkhanloo F, Ghobadi K (2020b) An open-source dataset on dietary behaviors and dash eating plan optimization constraints. Working paper.

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

Akhtar SA, Kolarijani AS, Esfahani PM (2022) Learning for control: An inverse optimization approach. *IEEE Control Systems Lett.* 6:187–192.

Aswani A, Kaminsky P, Mintz Y, Flowers E, Fukuoka Y (2019) Behavioral modeling in weight loss interventions. *Eur. J. Oper. Res.* 272(3):1058–1072.

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

Babier A, Boutilier JJ, Sharpe MB, McNiven AL, Chan TC (2018) Inverse optimization of objective function weights for treatment planning using clinical dose-volume histograms. *Phys. Medicine Biol.* 63(10):105004.

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, Martin A, Pokutta S, Schneider O (2018) An online-learning approach to inverse optimization. *arXiv preprint arXiv:1810.12997* .

Bastani H, Bastani O, Sinchaisri WP (2021) Learning best practices: Can machine learning improve human decision-making? Working paper.

Bazrafkan L, Choobineh MA, Shojaei M, Bozorgi A, Sharifi MH (2021) How do overweight people dropout of a weight loss diet? A qualitative study. *BMC Nutrition* 7(1):1–9.

Beam AL, Kohane IS (2018) Big data and machine learning in health care. *JAMA* 319(13):1317–1318.

Beil DR, Wein LM (2003) An inverse-optimization-based auction mechanism to support a multiattribute rfq process. *Management Sci.* 49(11):1529–1545.

Bengio Y, Lodi A, Prouvost A (2021) Machine learning for combinatorial optimization: A methodological tour d'horizon. *Eur. J. Oper. Res.* 290(2):405–421.

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

Birge JR, Li X, Sun C (2022) Stochastic inverse optimization. Working paper.

Burton D, Toint PL (1992) On an instance of the inverse shortest paths problem. *Math. Programming* 53(1-3):45–61.

Buttriss JL, Briend A, Darmon N, Ferguson EL, Maillot M, Lluch A (2014) Diet modelling: How it can inform the development of dietary recommendations and public health policy. *Nutrition Bull.* 39(1):115–125.

CDC (2020) Nhanes dietary data. URL https://wwwn.cdc.gov/nchs/nhanes/Search/DataPage.aspx?Component=Dietary.

Chakraborty S, Das S (2017) k-means clustering with a new divergence-based distance metric: Convergence and performance analysis. *Pattern Recognition Lett.* 100:67–73.

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

Chan TC, Eberg M, Forster K, Holloway C, Ieraci L, Shalaby Y, Yousefi N (2022a) An inverse optimization approach to measuring clinical pathway concordance. *Management Sci.* 68(3):1882–1903.

Chan TC, Forster K, Habbous S, Holloway C, Ieraci L, Shalaby Y, Yousefi N (2022b) Inverse optimization on hierarchical networks: An application to breast cancer clinical pathways. *Health Care Management Sci.* 1–33.

Chan TC, Mahmood R, Zhu IY (2021) Inverse optimization: Theory and applications. *arXiv preprint arXiv:2109.03920* .

Chan TCY, Lee T, Terekhov D (2019) Inverse optimization: Closed-form solutions, geometry, and goodness of fit. *Management Sci.* 65(3):1115–1135.

Chen N, Hu M, Li W (2022) Algorithmic decision-making safeguarded by human knowledge. *arXiv preprint arXiv:2211.11028* .

Chow JY, Recker WW (2012) Inverse optimization with endogenous arrival time constraints to calibrate the household activity pattern problem. *Transportation Res. Part B: Methodological* 46(3):463–479.

Dai T, Singh S (2020) Conspicuous by its absence: Diagnostic expert testing under uncertainty. *Marketing Science* 39(3):540–563.

Dai T, Singh S (2022) Artificial intelligence on call: The physician's decision of whether to use AI in clinical practice. Working paper.

Dai T, Tayur S (2020) Healthcare operations management: A snapshot of emerging research. *Manufacturing Service Oper. Management* 22(5):869–887.

Dantzig GB (1965) *Linear Programming and Extensions*, volume 48 (Princeton, NJ: Princeton University Press).

DASH (2018) Dash eating plan. URL https://www.nhlbi.nih.gov/health-topics/dash-eating-plan.

De Raedt L, Passerini A, Teso S (2018) Learning constraints from examples. *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, 7965–7970, AAAI'18/IAAI'18/EAAI'18.

de Verícourt F, Gurkan H (2022) Is your machine better than you? You may never know. Working paper.

Dietvorst BJ, Simmons JP, Massey C (2015) Algorithm aversion: people erroneously avoid algorithms after seeing them err. *J. Experiment. Psych. General* 144(1):114.

Downer MK, Gea A, Stampfer M, Sánchez-Tainta A, Corella D, Salas-Salvadó J, Ros E, Estruch R, Fitó M, Gómez-Gracia E, et al. (2016) Predictors of short-and long-term adherence with a mediterranean-type diet intervention: the predimed randomized trial. *Internat. J. Behav. Nutrition Physical Activity* 13(1):1–16.

Elmachtoub AN, Grigas P (2022) Smart "predict, then optimize". *Management Sci.* 68(1):9–26.

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

Faragó A, Szentesi Á, Szviatovszki B (2003) Inverse optimization in high-speed networks. *Discrete Applied Mathematics* 129(1):83–98.

Firdaus H, Hassan SI, Kaur H (2018) A comparative survey of machine learning and meta-heuristic optimization algorithms for sustainable and smart healthcare. *African J. Comput. ICT Ref. Format* 11(4):1–17.

Garille SG, Gass SI (2001) Stigler's diet problem revisited. *Oper. Res.* 49(1):1–13.

Gazan R, Brouzes CM, Vieux F, Maillot M, Lluch A, Darmon N (2018) Mathematical optimization to explore tomorrow's sustainable diets: A narrative review. *Adv. Nutrition* 9(5):602–616.

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

Ghobadi K, Mahmoudzadeh H (2020) Inferring linear feasible regions using inverse optimization. Working paper.

Goldenberg SL, Nir G, Salcudean SE (2019) A new era: artificial intelligence and machine learning in prostate cancer. *Nature Rev. Urology* 16(7):391–403.

Grand-Clément J, Pauphilet J (2022) The best decisions are not the best advice: Making adherence-aware recommendations. Working paper.

Gupte A, Ahmed S, Cheon MS, Dey S (2013) Solving mixed integer bilinear problems using milp formulations. *SIAM J. Optim.* 23(2):721–744.

Ibrahim R, Kim SH, Tong J (2021) Eliciting human judgment for prediction algorithms. *Management Sci.* 67(4):2314–2325.

Inelmen EM, Toffanello ED, Enzi G, Gasparini G, Miotto F, Sergi G, Busetto L (2005) Predictors of drop-out in overweight and obese outpatients. *Internat. J. Obesity* 29(1):122–128.

Irz X, Leroy P, Réquillart V, Soler LG (2016) Beyond wishful thinking: Integrating consumer preferences in the assessment of dietary recommendations. *PLOS One* 11(6):e0158453.

Ivancic A, Kanellopoulos A, Geleijnse JM (2020) Diet modelling: Combining mathematical programming models with data-driven methods. *IFIP Advances in Information and Communication Technology*, 72–80 (Wageningen, The Netherlands: Springer International Publishing).

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

Keskinocak P, Savva N (2020) A review of the healthcare-management (modeling) literature published in manufacturing & service operations management. *Manufacturing Service Oper. Management* 22(1):59–72.

Lin W, Kim SH, Tong J (2022) Does algorithm aversion exist in the field? An empirical analysis of algorithm use determinants in diabetes self-management. Working paper.

Longoni C, Bonezzi A, Morewedge CK (2019) Resistance to medical artificial intelligence. *J. Consumer Res.* 46(4):629–650.

Maillot M, Vieux F, Amiot MJ, Darmon N (2010) Individual diet modeling translates nutrient recommendations into realistic and individual-specific food choices. *Amer. J. Clinical Nutrition* 91(2):421–430.

Márquez-Neila P, Salzmann M, Fua P (2017) Imposing hard constraints on deep networks: Promises and limitations. *arXiv preprint arXiv:1706.02025* .

Misra S, Roald L, Ng Y (2018) Learning for constrained optimization: Identifying optimal active constraint sets. *arXiv preprint arXiv:1802.09639* .

Morgenstern JD, Rosella LC, Costa AP, de Souza RJ, Anderson LN (2021) Perspective: big data and machine learning could help advance nutritional epidemiology. *Adv. Nutrition* 12(3):621–631.

Rodgers GP, Collins FS (2020) Precision nutrition—the answer to "what to eat to stay healthy". *JAMA* 324(8):735.

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

Selim SZ, Ismail MA (1984) K-means-type algorithms: A generalized convergence theorem and characterization of local optimality. *IEEE Trans. Pattern Anal. Machine Intelligence* (1):81–87.

Shahmoradi Z, Lee T (2022a) Optimality-based clustering: An inverse optimization approach. *Oper. Res. Lett.* 50(2):205–212.

Shahmoradi Z, Lee T (2022b) Quantile inverse optimization: Improving stability in inverse linear programming. *Oper. Res.* 70(4):2538–2562.

Stigler GJ (1945) The cost of subsistence. *J. Farm Econom.* 27(2):303–314.

Suen S, Negoescu D, Goh J (2022) Design of incentive programs for optimal medication adherence in the presence of observable consumption. *Oper. Res.* 70(3):1691–1716.

Terwiesch C, Olivares M, Staats BR, Gaur V (2020) OM Forum—A review of empirical operations management over the last two decades. *Manufacturing Service Oper. Management* 22(4):656–668.

Waring J, Lindvall C, Umeton R (2020) Automated machine learning: Review of the state-of-the-art and opportunities for healthcare. *Artificial Intelligence in Medicine* 104:101822.

Zhang J, Ma Z (1999) Solution structure of some inverse combinatorial optimization problems. *J. Combin. Optim.* 3(1):127–139.