Inverse Linear Optimization with Interval Coefficients

M. Hladík[1] and M. Černý[2]

[1]Department of Applied Mathematics, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic
[2]Department of Econometrics, University of Economics, Prague, Czech Republic

ICPAM-LAE 2013

Part I. Introduction

- What is optimization?
- Applications in Operations Research, Economics, Statistics, Game Theory, ...
- Inverse optimization

What is optimization?

- **In mathematics:** optimization is a theory on maximization or minimization of functions defined on special sets.
  - We are given an **objective function** $f : \mathbb{R}^n \to \mathbb{R}$ which is to be maximized/minimized over a given region $\Omega \subseteq \mathbb{R}^n$, called **feasible region**.
- **In applications:** we are always maximizing or minimizing something, for example:
  - **Microeconomics:** we want to maximize profit of a firm or minimize its costs under limited resources;
  - **Statistics:** we want to maximize likelihood or minimize residual error (e.g. residual sum of squares);
  - **Experimental Design:** we want to maximize measurement precision in an experiment under restricted possibilities of available laboratory equipment;
  - **Portfolio Theory:** we want to maximize return of an investment under budget constraints and regulatory constraints;
  - **Operations Research:** we want to minimize the length of the path of the Traveling Salesman;
  - many more applications in engineering, physics, chemistry etc.

A basic example: Nutrition (Diet) Problem

In practice, this problem is solved e.g. by producers of dog food, often on a daily basis.

- A producer of dog food processes leftovers from butchers, slaughterhouses and meat-processing plants.
- The producer must combine the available raw materials to achieve the declared nutrition content, e.g. enough proteins, enough calories, not too much salt, not too much fat etc.
- The producer does not care about what the ingredients exactly are:
  - (s)he simply buys anything from which it is possible to combine the declared nutrition levels, as cheaply as possible,
  - (s)he mixes and boils the raw materials, getting a homogenous tasteless mesh,
  - (s)he adds meat perfume,
  - (s)he fills the 'product' into cans,
  - (s)he adds one piece of real meat just under the cover of each can (for a better visual effect),
  - (s)he spends plenty of money on marketing to be able to sell this stuff.

A basic example: Nutrition Problem (continued)

For example assume that a meat processing plant offers two kinds of leftovers:

- $x_1 =$ leftover of salami,
- $x_2 =$ mechanically separated meat.

Now we can summarize data for our problem: we know the contents of proteins, fat, and salt in each of the two ingredients x_1, x_2, and we know their prices per ton.

Our goal. We want to mix them to obtain dog food containing at least a declared level of proteins, at most a declared level of fat and at most a declared level of salt. Furthermore, we want to minimize costs.

A basic example: Nutrition Problem (continued)

We get the following optimization problem:

	salami		sep.meat		demand
minimize	$c_1 x_1$	$+$	$c_2 x_2$		
subject to:					
proteins	$a_{11} x_1$	$+$	$a_{12} x_2$	$\geqslant$	b_1
fat	$a_{21} x_1$	$+$	$a_{22} x_2$	$\leqslant$	b_2
salt	$a_{31} x_1$	$+$	$a_{32} x_2$	$\leqslant$	b_3
	x_1			$\geqslant$	0
			x_2	$\geqslant$	0

Data of the optimization problem are denoted in red:

- a_{ij} denote the contents of proteins, fat, salt in one tone of salami and separated meat,
- b_i denote the demands,
- c_j denote the prices per ton of salami and separated meat.

Examples of optimization problems — nonlinear case

- Portfolio optimization. Data: C, r, γ. Find a portfolio with average yield $\geqslant \gamma$ and minimal variance:

$$\min_x x^{\mathrm{T}} C x \quad \text{s.t.} \quad \sum_i x_i = 1, x \geqslant 0, \sum_i r_i x_i \geqslant \gamma.$$

- Logistic regression. Data: observed pairs $[x_1, y_1], \ldots, [x_n, y_n]$. Find least-squares estimates of regression parameters $\beta_1, \beta_2, \beta_3$:

$$\min_{\beta_1,\beta_2,\beta_3} \sum_{i=1}^{n} \left[y_i - \frac{\beta_1}{1 + e^{-\beta_2(x_i - \beta_3)}} \right]^2 \quad \text{s.t.} \quad \beta_1 \geqslant 0, \beta_2 \geqslant 0.$$

- Smallest-volume circumscribing ellipsoid. Data: points $x_1, \ldots, x_n \in \mathbb{R}^n$. Solve

$$\min_{E \in \mathbb{R}^{n\times n}, s \in \mathbb{R}^n} \det E \quad \text{s.t.} \quad (\forall i)\ (x_i - s)^{\mathrm{T}} E^{-1} (x_i - s) \leqslant 1,\ \ E \text{ p.s.d.}$$

Part II. Inverse optimization

- General formulation
- Questions and problems
- Optimal value function
- Applications

General formulation

Let $\Theta \subseteq \mathbb{R}^k$ (admissible region) be given. Consider the class of optimization problems

$$\min_x \{\varphi(x;\theta) : \; g_1(x;\theta) \leqslant 0, \; \ldots, \; g_m(x;\theta) \leqslant 0, \; x \in \mathbb{R}^n\} \quad \theta \in \Theta, \qquad (1)$$

where

- $x \in \mathbb{R}^n$: vector of variables,
- $\theta \in \mathbb{R}^k$: data vector,
- $\varphi : \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}$: objective function,
- $g_1, \ldots, g_m : \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}$: constraint functions.

When we fix $\theta_0 \in \Theta$, we say that we select the scenario θ_0, or, that we select the optimization problem

$$\min_x \varphi(x;\theta_0) \;\; \text{s.t.} \;\; g_1(x;\theta_0) \leqslant 0, \; \ldots, \; g_m(x;\theta_0) \leqslant 0$$

from the family (1).

Inverse optimization problem: given $\lambda_0 \in \mathbb{R}$, find $\theta_0 \in \Theta$ such that

$$\min_x \{\varphi(x;\theta_0) : \; g_1(x;\theta_0) \leqslant 0, \; \dots, \; g_m(x;\theta_0) \leqslant 0\} = \lambda_0$$

or assert that none exists.

Interpretation. The data of the inverse optimization problem consist of the functions $\varphi, g_1, \dots, g_m$, the set Θ and the value λ_0, called demand. We can say that we are to 'design' an optimization problem

$$\min_x \{\varphi(x;\theta_0) : g_1(x;\theta_0) \leqslant 0, \dots, g_m(x;\theta_0) \leqslant 0\}$$

(or: 'select a scenario') attaining the prescribed optimal value λ_0. Our constraints are that we are allowed to select the parameter vector θ_0 only from the admissible set Θ.

Special case: Inverse linear programming

- **Traditional linear programming.**
  - **Data:** $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$.
  - **Task:** Find $x \in \mathbb{R}^n$ solving $\min c^{\mathrm{T}}x$ s.t. $Ax = b, x \geqslant 0$.
- **Inverse linear programming.**
  - **Data:** $\Theta \subseteq \mathbb{R}^{m\times n} \times \mathbb{R}^m \times \mathbb{R}^n$ and $\lambda_0 \in \mathbb{R}$.
  - **Task:** Find $(A, b, c) \in \Theta$ such that $\min\{c^{\mathrm{T}}x : \ Ax = b, x \geqslant 0\} = \lambda_0$.
- **Inverse linear programming with interval coefficients.**
  - **Data:** $\mathbf{A} \in \mathbb{IR}^{m\times n}$, $\mathbf{b} \in \mathbb{IR}^m$, $\mathbf{c} \in \mathbb{IR}^n$ and $\lambda_0 \in \mathbb{R}$.
  - **Task:** Find $(A, b, c) \in \Theta := (\mathbf{A}, \mathbf{b}, \mathbf{c})$ such that $\min\{c^{\mathrm{T}}x : \ Ax = b, x \geqslant 0\} = \lambda_0$.
  - Here: $\mathbb{IR}^{m\times n}$ is the space of all interval matrices. An interval matrix is a family of matrices

$$\mathbf{A} = [\underline{A}, \overline{A}] = \{A \in \mathbb{R}^{m\times n} : \underline{A} \leqslant A \leqslant \overline{A}\},$$

where "$\leqslant$" is understood componentwise.

Questions and problems

From now on: we will study only the case of inverse LP with interval coefficients ("IILP"). Of course, the questions and problems apply more generally.

Solution space. Let Θ^* denote the set of solutions to IILP, i.e.

$$\Theta^* = \big\{(A,b,c) \in (\mathbf{A},\mathbf{b},\mathbf{c}) : \min_x \{c^{\mathrm{T}} x : Ax = b, x \geqslant 0\} = \lambda_0\big\}.$$

Some natural questions.

- Can we test whether $\Theta^* \neq \emptyset$? (I.e., can we test whether the problem has at least one solution?) Can we decide by an efficient algorithm (i.e., in polynomial time), or is the problem computationally hard (say, NP-hard)?
- Can we test whether Θ^* is a singleton? (I.e., can we test uniqueness of the solution?)
- How to find some $\theta^* \in \Theta^*$?
- How to describe or approximate the set Θ^* if it is intricate?
- How to determine further set-theoretic properties of Θ^*, such as connectivity, (un)boundedness etc.?

Questions and problems (contd.)

Optimal value function:

$$f(A,b,c) = \inf_x \{c^{\mathrm{T}}x : Ax = b, x \geqslant 0\}.$$

Remark.

- $f(A,b,c) = -\infty$ means that the scenario (A,b,c) is unbounded.
- $f(A,b,c) = \infty$ means that the scenario (A,b,c) is infeasible.

Some natural questions:

- Is the optimal value function f continuous on $\mathbf{A} \times \mathbf{b} \times \mathbf{c}$?
- Is the optimal value function f monotone on $\mathbf{A} \times \mathbf{b} \times \mathbf{c}$?
- Does it hold $f(A,b,c) = \infty$ for some A,b,c?
- Does it hold $f(A,b,c) = -\infty$ for some A,b,c?
- More generally: how to describe the range of $f(A,b,c)$ over the domain $\mathbf{A} \times \mathbf{b} \times \mathbf{c}$?

Two applications

Example 1. Finding a maximal flow in a network can be written as a linear programming problem.

- Data: capacities of edges.
- IILP: given possible intervals for capacities of edges, find the capacities in order to achieve the prescribed maximal flow.

Example 2. Designing a matrix game.

- Data: payoff matrix.
- IILP: given an interval of admissible payoff matrices, find the payoff matrix with a prescribed value of the game.
- Remark. Recall that finding the Nash mixed strategy can be solved via the linear program

$$\max_{\gamma,x} \gamma \quad \text{s.t.} \quad Ax \geqslant \gamma e, x \geqslant 0, e^{\mathrm{T}} x = 1,$$

where $e = (1, \ldots, 1)^{\mathrm{T}}$ and A is the payoff matrix.

Part III. Some theory of IILP

- Continuity of the optimal value function
- Binary Search
- Parametric programming

Some theory of IILP

Two observations:

- The optimal value function $f(A, b, c) = \min\{c^{\mathrm{T}}x : Ax = b, x \geqslant 0\}$ is computable in polynomial time ("easy-to-evaluate"), using e.g. Interior Point Methods.
- The admissible space $\Theta = \mathbf{A} \times \mathbf{b} \times \mathbf{c}$ is a convex set.

Crucial questions:

- Is the optimal value function f continuous?
- Given λ_0,
  - (lower bound): can we find a scenario $(A_0, b_0, c_0) \in (\mathbf{A}, \mathbf{b}, \mathbf{c})$ such that $f(A_0, b_0, c_0) < \lambda_0$?
  - (upper bound): can we find a scenario $(A_1, b_1, c_1) \in (\mathbf{A}, \mathbf{b}, \mathbf{c})$ such that $f(A_1, b_1, c_1) > \lambda_0$?

If all answers are positive, we can use Binary Search: by convexity of $\mathbf{A} \times \mathbf{b} \times \mathbf{c}$ we can define

$$v(\mu) = f((1-\mu)A_0 + \mu A_1, (1-\mu)b_0 + \mu b_1, (1-\mu)c_0 + \mu c_1) - \lambda_0$$

and using fast computability we simply find its root over $\mu \in [0, 1]$.

Binary search

To recall:

$$v(\mu) = f((1-\mu)A_0 + \mu A_1, (1-\mu)b_0 + \mu b_1, (1-\mu)c_0 + \mu c_1) - \lambda_0.$$

Binary Search:

(1) input: precision parameter $\varepsilon > 0$, λ_0, (A_0, b_0, c_0), (A_1, b_1, c_1).

(2) set $\underline{\mu} := 0$, $\overline{\mu} := 1$

(3) set $\mu' := \frac{1}{2}(\underline{\mu} + \overline{\mu})$

(4) if $|v(\mu') - \lambda_0| < \varepsilon$ then return the scenario $((1-\mu')A_0 + \mu' A_1, (1-\mu')b_0 + \mu' b_1, (1-\mu')c_0 + \mu' c_1)$ and terminate

(5) if $v(\mu') < \lambda_0$ then set $\underline{\mu} := \mu'$

(6) if $v(\mu') > \lambda_0$ then set $\overline{\mu} := \mu'$

(7) go to 3.

Is the optimal value function continuous?

Problem: the optimal value function
$f(A,b,c) = \min\{c^{\mathrm{T}}x : Ax = b, x \geqslant 0\}$ need not be continuous.

Theorem. If every scenario $(A,b,c) \in (\mathbf{A},\mathbf{b},\mathbf{c})$ satisfies

(a) $\{x : Ax = 0, x \geqslant 0, c^{\mathrm{T}}x \leqslant 0\} = \{0\}$,

(b) $\{y : A^{\mathrm{T}}y \leqslant 0, b^{\mathrm{T}}y \geqslant 0\} = \{0\}$,

then f is continuous.

Theorem. The condition (a) is satisfied iff the linear programming problem

$$\underline{A}x \leqslant 0,\ \overline{A}x \geqslant 0,\ x \geqslant 0,\ \underline{c}^{\mathrm{T}}x \leqslant 0,\ \sum_i x_i = 1$$

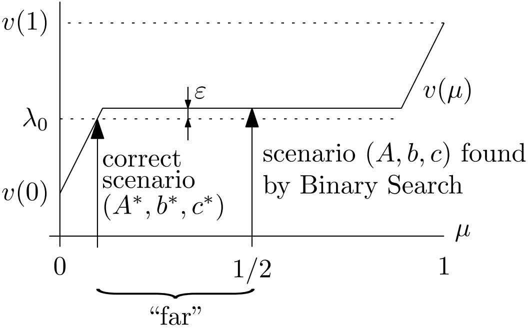
is infeasible.

Corollary. Testing whether (a) holds can be done in polynomial computation time.

Unfortunately: Testing (b) is NP-hard. But this would be another story...

A drawback of Binary Search

Binary Search finds the desired scenario only approximately (depending on the precision parameter $\varepsilon > 0$), meaning that it can make a significant error:



Parametric programming technique

To recall: an index set $B = \{i_1, \dots, i_n\}$ is a basis if A_B is nonsingular. (If a_i is i-th column of A, then A_B has columns $a_{i_1}, \dots, a_{i_n}$.)

Roughly: a basis B is optimal if $x = A_B^{-1} b$ is an optimal solution of the linear programming problem $\min\{c^{\mathrm{T}} x : Ax = b, x \geqslant 0\}$.

We denote $R := \{1, \dots, m\} \setminus B$ and write A_R accordingly.

A well-known theorem from LP. A basis B is optimal iff the following conditions hold:

- feasibility condition: $A_B^{-1} b \geqslant 0$,
- optimality condition: $c_R - c_B A_B^{-1} A_R \geqslant 0$.

Parametric programming technique (contd.)

The main ingredient — "a shift from scenario $A(\kappa_0), b(\kappa_0), c(\kappa_0)$ to scenario $A(\kappa_1), b(\kappa_1), c(\kappa_1)$".

- Assume that A, b, c depend on a parameter κ and write $A(\kappa), b(\kappa), c(\kappa)$.
- Let $\kappa_0 < \kappa_1$ be given. Let B be an optimal basis for $\min\{c(\kappa_0)^{\mathrm{T}}x : A(\kappa_0)x = b(\kappa_0), x \geqslant 0\}$.
- Find

$$\kappa^* = \min\{\kappa_1, \sup\{\kappa : B \text{ is an optimal basis for } \min\{c(\kappa)^{\mathrm{T}}x : A(\kappa)x = b(\kappa), x \geqslant 0\}\}\}. \quad (2)$$

  (Remark: we find the maximal κ such that both the feasibility condition $A(\kappa)_B^{-1}b(\kappa) \geqslant 0$ and the optimality condition $c(\kappa)_R - c(\kappa)_B A(\kappa)_B^{-1}A(\kappa)_R \geqslant 0$ hold.)
- If $\kappa^* < \kappa_1$, then $\min\{c(\kappa^*)^{\mathrm{T}}x : A(\kappa^*)x = b(\kappa^*), x \geqslant 0\}$ must have another optimal basis B. So, find it and repeat (2). Stop when $\kappa^* = \kappa_1$.

Parametric programming technique (contd.)

Crucial issue: how to find $\kappa^* = \sup\{\kappa : B$ is an optimal basis for $\min\{c(\kappa)^{\mathrm{T}}x : A(\kappa)x = b(\kappa), x \geqslant 0\}\}$?

Without details: we have the following theorem.

- If $A(\kappa) = A$, $b(\kappa) = b$ and $c(\kappa) = (1-\kappa)c_0 + \kappa c_1$, then κ^* can be found in polynomial time.
- If $A(\kappa) = A$, $b(\kappa) = (1-\kappa)b_0 + \kappa b_1$ and $c(\kappa) = c$, then κ^* can be found in polynomial time.
- Rank-one lemma: If $A(\kappa) = A_0 + \kappa A^*$, where A^* has rank one, $b(\kappa) = b$ and $c(\kappa) = c$, then κ^* can be found in polynomial time.

Parametric programming technique (contd.)

Now we can roughly describe the parametric programming method.

- Let the optimal value function be continuous.
- Let (A_0, b_0, c_0) be a scenario s.t. $\min\{c_0^{\mathrm{T}} x : A_0 x = b_0, x \geqslant 0\} < \lambda_0$.
- Let (A_1, b_1, c_1) be a scenario s.t. $\min\{c_1^{\mathrm{T}} x : A_1 x = b_1, x \geqslant 0\} > \lambda_0$.
- Stage I. Set $A = A_0$, $b = b_0$, $c = (1-\kappa)c_0 + \kappa c_1$ and shift $(A_0, b_0, c_0) \to (A_0, b_0, c_1)$.
- Stage II. Set $A = A_0$, $b = (1-\kappa)b_0 + \kappa b_1$, $c = c_1$ and shift $(A_0, b_0, c_1) \to (A_0, b_1, c_1)$.
- Stage III.
  - Choose a rank-one decomposition: choose a sequence of rank-one matrices $A_1^*, \ldots, A_\ell^*$ such that $A_0 + A_1^* + \cdots, A_\ell^* = A_1$.
  - Make shifts

$$(A_0, b_1, c_1) \to (A_0 + A_1^*, b_1, c_1) \to (A_0 + A_1^* + A_2^*, b_1, c_1) \to$$
$$\to \cdots \to (A_0 + A_1^* + A_2^* + \cdots + A_\ell^*, b_1, c_1) = (A_1, b_1, c_1).$$

Conclusions: pros and cons

Binary Search:

- $(+)$ in practice: often fast
- $(-)$ only ε-exact solution is found
- $(-)$ it is hard to find a theoretical bound on the number of iterations to achieve ε-convergence

Parametric Programming Approach:

- $(-)$ in practice: usually slower than Binary Search
- $(+)$ exact solution is found
- $(+)$ more flexible (a user can choose what will be perturbed first)
- $(-)$ worst-case complexity can be bad (similar to the Simplex Algorithm)

Both approaches:

- $(-)$ continuity of the optimal value function is required
- $(-)$ a-priori knowledge of (A_0, b_0, c_0) and (A_1, b_1, c_1) is required

Thank You! And... some further reading...

M. Fiedler, J. Nedoma, J. Ramík, J. Rohn, and K. Zimmermann.
Linear optimization problems with inexact data.
Springer, New York, 2006.

M. Hladík.
Interval linear programming: A survey.
In Z. A. Mann, editor, *Linear Programming - New Frontiers in Theory and Applications*, chapter 2, pages 85–120. Nova Sci. Pub., 2012.

M. Hladík and M. Černý.
Interval regression by tolerance analysis approach.
Fuzzy Sets Syst., 193:85–107, 2012.

A. Mostafaee, M. Hladík, and M. Černý.
Inverse linear programming problem with interval coefficients.
unpublished, 2013.

M. Černý, J. Antoch, and M. Hladík.
On the possibilistic approach to linear regression models involving uncertain, indeterminate or interval data.
Inf. Sci., 244:26–47, 2013.