An interval linear programming contractor

Milan Hladík[1]

Abstract. We consider linear programming with interval data. One of the most challenging problems in this topic is to determine or tight approximate the set of all optimal solutions subject to all perturbations within the given intervals. We propose an iterative method that finds an enclosure of the set of optimal solutions. The method is based on a linear approximation and sequential refinement. It runs in polynomial time, so, naturally, convergence to the ideal set cannot be ensured. We apply the method in a simple portfolio selection problem with uncertain data.

Keywords: interval linear programming, interval analysis

JEL classification: C44
AMS classification: 90C31

1 Introduction

Since linear programming is widely used in modelling real-life problems, it must take into account inaccuracies and data measurement errors, which are common in most of the problems. There are plenty of various approaches to handle uncertainty in linear systems, see e.g. [4]. In this paper, we deal with an interval linear programming, in which we assume that there are *a priori* known some intervals in which inexact quantities may perturb. Linear programming with interval data has been studied for forty years; see a survey paper [10]. The problems discussed are the optimal value range [4], [7], [12], basis stability, and duality [5], among others. Interval linear programming was applied in portfolio selection problems [6], environmental management [15] interval matrix games [16], and can also serve in fuzzy linear regression as an alternative to traditional approaches [11, 3].

In this paper, we focus on the optimal solution set. The problem of calculating the set of all possible solutions over all data perturbations is considered to be very difficult. It becomes tractable in the special case of the so called basis stability [8], [14], [17], meaning that there is a basis that is optimal under any admissible perturbation. However, basis stability is not so easy to verify; indeed, it is an NP-hard problem [10]. Moreover, since many practical problems suffer from degeneracy, one cannot expect that basis stability holds true in general.

Thus, the research was driven to calculate an enclosure (interval superset) of the optimal solution set. Such enclosures can be computed e.g. by using interval arithmetic [2], [13], instead of the real one, but the results are usually very overestimated.

Our aim is to propose an efficient algorithm for computing an enclosure of the optimal solution set. We present an iterative algorithm that starts with an initial enclosure and sequentially makes it tighter. Naturally, it doesn't converge to the optimal enclosure in general, but (in view of the performed examples) it gives bounds that are sufficiently accurate for many purposes.

Let us introduce some notations. An interval matrix is a family of matrices

$$\boldsymbol{A} := \{A \in \mathbb{R}^{m\times n} \mid \underline{A} \le A \le \overline{A}\},$$

where $\underline{A}, \overline{A} \in \mathbb{R}^{m\times n}$ are given. The midpoint and the radius of an interval matrix $\boldsymbol{A}$ is defined respectively as

$$A^c := \frac{1}{2}(\underline{A} + \overline{A}), \quad A^\Delta := \frac{1}{2}(\overline{A} - \underline{A}),$$

[1]Charles University, Faculty of Mathematics and Physics, Department of Applied Mathematics, Malostranské nám. 25, 118 00, Prague, Czech Republic, and
University of Economics, Faculty of Informatics and Statistics, nám. W. Churchilla 4, 13067, Prague, Czech Republic, e-mail: milan.hladik@matfyz.cz

and similarly for interval vectors. The set of all interval $m \times n$ matrices is denoted by $\mathbb{IR}^{m\times n}$. A solution to the interval system $\boldsymbol{A}x = \boldsymbol{b}$, $x \geq 0$ means any solution to any scenario $Ax = b$, $x \geq 0$ with $A \in \boldsymbol{A}$, $b \in \boldsymbol{b}$. Similarly for interval systems of inequalities. The diagonal matrix with entries $z_1, \dots, z_n$ is denoted by $\operatorname{diag}(z)$.

2 Interval linear programming

Consider a linear programming problem

$$\min c^T x \ \text{ subject to } \ Ax = b,\ x \geq 0, \tag{1}$$

and its dual

$$\max b^T y \ \text{ subject to } \ A^T y \leq c.$$

Suppose that (possibly all) input values are inexact, and we have lower and upper limits for the ranges of variations. Thus, we are given $\boldsymbol{A} \in \mathbb{IR}^{m\times n}$, $\boldsymbol{b} \in \mathbb{IR}^m$, $\boldsymbol{c} \in \mathbb{IR}^n$, and we focus on the family of linear programs (1) with $A \in \boldsymbol{A}$, $b \in \boldsymbol{b}$, $c \in \boldsymbol{c}$. Denote by $\mathcal{S}(A, b, c)$ the set of optimal solutions to (1). By the set of optimal solutions to an interval linear program we understand the set

$$\mathcal{S} := \bigcup_{A\in\boldsymbol{A},\, b\in\boldsymbol{b},\, c\in\boldsymbol{c}} \mathcal{S}(A, b, c).$$

The set $\mathcal{S}$ needn't be a polyhedron, and it is hard to determine it in general. We want to give a cheap calculation for an enclosure to $\mathcal{S}$.

By using duality theory, we have that $x \in \mathcal{S}$ if and only if there is some $y \in \mathbb{R}^m$, $A \in \boldsymbol{A}$, $b \in \boldsymbol{b}$, and $c \in \boldsymbol{c}$ such that

$$Ax = b,\ x \geq 0,\ A^T y \leq c,\ c^T x = b^T y. \tag{2}$$

Relaxing the dependency we obtain a superset described by

$$\boldsymbol{A}x = \boldsymbol{b},\ x \geq 0,\ \boldsymbol{A}^T y \leq \boldsymbol{c},\ \boldsymbol{c}^T x = \boldsymbol{b}^T y. \tag{3}$$

Notice that the solution sets are not equal, since in the latter the intervals vary independently within $\boldsymbol{A}$ and $\boldsymbol{A}^T$, while in the former they are related. By [10], the solution set to (3) is described as

$$\underline{A}x \leq \overline{b},\ -\overline{A}x \leq -\underline{b},\ x \geq 0,\ A_c^T y - A_\Delta^T |y| \leq \overline{c},\ |c_c^T x - b_c^T y| \leq c_\Delta^T x + b_\Delta^T |y|. \tag{4}$$

It represents a non-convex polyhedron, which becomes convex when we restrict the signs of y_i, $i = 1, \dots, m$. A simple method introduced in [10] was based on decomposing (4) into 2^m sub-problems according to the signs of y_i, $i = 1, \dots, m$. Each sub-problem has a linear description and is solved by ordinary linear programming. However, the method is very time consuming as m grows.

In the following, we employ a kind of linearization of (4), which is based on a result by Beaumont [1]. It needs an initial enclosure, which is used for refinement.

Theorem 1 (Beaumont, 1998)**.** *For every $y \in \boldsymbol{y} \subset \mathbb{R}$ with $\underline{y} < \overline{y}$ one has*

$$|y| \leq \alpha y + \beta, \tag{5}$$

where

$$\alpha = \frac{|\overline{y}| - |\underline{y}|}{\overline{y} - \underline{y}} \ \textit{ and } \ \beta = \frac{\overline{y}|\underline{y}| - \underline{y}|\overline{y}|}{\overline{y} - \underline{y}}.$$

Moreover, if $\underline{y} \geq 0$ or $\overline{y} \leq 0$ then (5) *holds as equation.*

Let $\boldsymbol{x} \in \mathbb{IR}^n$ and $\boldsymbol{y} \in \mathbb{IR}^m$ be an enclosure to (4). By the Beaumont theorem, we will linearize the absolute value $|y|$ as follows. Define vectors $\alpha, \beta \in \mathbb{R}^m$ componentwise as

$$\alpha_i := \begin{cases} \frac{|\overline{y}_i| - |\underline{y}_i|}{\overline{y}_i - \underline{y}_i} & \text{if } \ \underline{y}_i < \overline{y}_i, \\ \operatorname{sgn}(\overline{y}_i) & \text{if } \ \underline{y}_i = \overline{y}_i, \end{cases}$$

$$\beta_i := \begin{cases} \frac{\overline{y}_i|\underline{y}_i| - \underline{y}_i|\overline{y}_i|}{\overline{y}_i - \underline{y}_i} & \text{if } \ \underline{y}_i < \overline{y}_i, \\ 0 & \text{if } \ \underline{y}_i = \overline{y}_i. \end{cases}$$

Then the linearization of (4) reads

$$\underline{A}x \le \overline{b},\ -\overline{A}x \le -\underline{b},\ x \ge 0, \tag{6a}$$

$$\big(A_c^T - A_\Delta^T \operatorname{diag}(\alpha)\big)y \le \overline{c} + A_\Delta^T \beta, \tag{6b}$$

$$\underline{c}^T x + \big(-b_c^T - b_\Delta^T \operatorname{diag}(\alpha)\big)y \le b_\Delta^T \beta, \tag{6c}$$

$$-\overline{c}^T x + \big(b_c^T - b_\Delta^T \operatorname{diag}(\alpha)\big)y \le b_\Delta^T \beta. \tag{6d}$$

Now, we compute the interval hull to (6). If it is smaller than the initial enclosure $\boldsymbol{x}, \boldsymbol{y}$ then we can iterate the process to obtain more tight enclosure. It is a basic idea of our method described in Algorithm 1.

The initial enclosure $\boldsymbol{x}^0, \boldsymbol{y}^0$ from step 1 is taken as

$$\boldsymbol{x}^0 := ([0, K], \dots, [0, K])^T, \quad \boldsymbol{y}^0 := ([-K, K], \dots, [-K, K])^T,$$

where $K \gg 0$ is large enough. The stopping criterion used in step 7 is

$$\frac{\sum_{j=1}^n (x_\Delta^i)_j + \sum_{j=1}^m (y_\Delta^i)_j}{\sum_{j=1}^n (x_\Delta^{i-1})_j + \sum_{j=1}^m (y_\Delta^{i-1})_j} \ge 0.99.$$

Algorithm 1 (Optimal solution set contractor)

1: Compute an initial interval enclosure $\boldsymbol{x}^0, \boldsymbol{y}^0$ of (4);
2: $i := 0$;
3: **repeat**
4: compute α and β by using $\boldsymbol{y}^i$;
5: $i := i + 1$;
6: compute the interval hull $\boldsymbol{x}^i, \boldsymbol{y}^i$ of (6);
7: **until** improvement is nonsignificant;
8: **return** $\boldsymbol{x}^i$;

Remark 1. There is a natural problem how to choose a sufficiently large K for the initial enclosure. Even though the algorithm works well for very conservative K (cf. Example 1), in some cases, we can validate that K is large enough.

Suppose that $\boldsymbol{x}^0, \boldsymbol{y}^0$ contain at least one optimal solution for some scenario (which is easy to satisfy). Next, suppose that (2) is solvable for each $A \in \boldsymbol{A}$, $b \in \boldsymbol{b}$, and $c \in \boldsymbol{c}$; this issue is discussed e.g. in [10]. Take the initial enclosure $\boldsymbol{x}^0, \boldsymbol{y}^0$ as

$$\boldsymbol{x}^0 := ([-1, K], \dots, [-1, K])^T, \quad \boldsymbol{y}^0 := ([-K, K], \dots, [-K, K])^T,$$

If the interval vectors $\boldsymbol{x}^1, \boldsymbol{y}^1$ from the next iteration are strictly inside the initial ones, that is, if $\underline{x}^0 < \underline{x}^1 \le \overline{x}^1 < \overline{x}^0$ and $\underline{y}^0 < \underline{y}^1 \le \overline{y}^1 < \overline{y}^0$, then $\boldsymbol{x}^1, \boldsymbol{y}^1$ comprise all optimal solutions (this is why we put the lower limits of $\boldsymbol{x}^0$ as -1 instead of 0). This is easily seen from the continuity reasons since, under our assumption, the optimal solution set is connected.

Example 1. Consider an interval linear program

$$\begin{aligned} \min -[15,16]x_1 - [17,18]x_2 \ \text{ subject to}& \\ x_1 &\le [10,11], \\ -x_1 + [5,6]x_2 &\le [25,26], \\ [6,6.5]x_1 + [3,4.5]x_2 &\le [81,82], \\ -x_1 &\le -1, \\ x_1 - [10,12]x_2 &\le -[1,2]. \end{aligned}$$

Even though it is not in the standard form (1), the associated primal–dual pair is the same as for (1). We take the initial enclosure

$$\begin{aligned} \boldsymbol{x}^0 &= 1000 \cdot ([-1,1],\ [-1,1])^T, \\ \boldsymbol{y}^0 &= 1000 \cdot ([0,1],\ [0,1],\ [0,1],\ [0,1],\ [0,1])^T. \end{aligned}$$

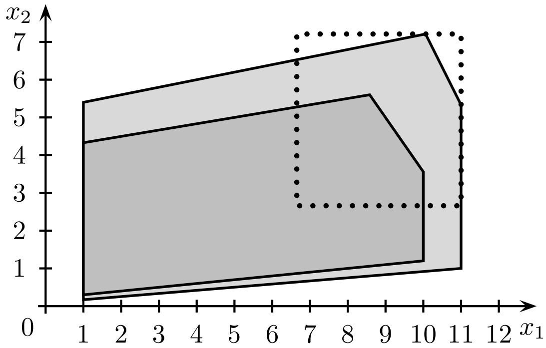

Figure 1: Intersection of all feasible sets in dark gray; union in light gray; the resulting enclosure of optimal solutions represented by the dotted rectangle.

The iterations of the procedure go as follows

$$\begin{aligned}
\boldsymbol{x}^1 &= ([1,11],\,[-568,916])^T,\\
\boldsymbol{y}^1 &= ([0,1000],\,[0,936],\,[0,358],\,[0,1000],\,[0,572])^T,\\
\boldsymbol{x}^2 &= ([1,11],\,[-17.2,72])^T,\\
\boldsymbol{y}^2 &= ([0,190],\,[0,58.5],\,[0,24.3],\,[0,176],\,[0,34.6])^T,\\
\boldsymbol{x}^3 &= ([3.78,11],\,[1.91,9.80])^T,\\
\boldsymbol{y}^3 &= ([0,30.6],\,[0,6.98],\,[4.71],\,[0,17.1],\,[0,3.09])^T,\\
\boldsymbol{x}^4 &= ([6.65,11],\,[2.66,7.21])^T,\\
\boldsymbol{y}^4 &= ([0,22.5],\,[0.08,4.33],\,[0,3.67],\,[0,8.81],\,[0,1.47])^T.
\end{aligned}$$

The fifth iteration does no improvement, so we return the enclosure $\boldsymbol{x}^4$. Notice that the same result is obtained by the exponential decomposition procedure discussed above, but with a much more computational effort. An illustration is given in Figure 1. We see that the upper bounds are very tight, but the lower bounds are quite conservative.

Example 2. Let us apply our approach in a simple portfolio selection model. Consider we have J possible investments for T time periods and r_{tj}, $t = 1,\dots,T$, $j = 1,\dots,J$, stands for return on investment j in time period t. Estimated reward on investment j using historical means is defined as

$$W_j := \frac{1}{T}\sum_{t=1}^{T} r_{tj}, \quad j = 1,\dots,J,$$

or in matrix form $W = \frac{1}{T}R^T e$, where e is the vector of ones (with convenient dimension). In order to get a linear programming formulation of the problem we measure risk of investment j by sum of absolute values instead of the sum of the squares:

$$\frac{1}{T}\sum_{t=1}^{T}|r_{tj} - W_j|.$$

Let μ be a risk aversion parameter (upper bound for risk) given by a user, and the variables x_j, $j = 1,\dots,J$, denotes a fraction of portfolio to invest in j. Then the maximal allowed risk is expressed by the constraint

$$\frac{1}{T}\sum_{t=1}^{T}\Big|\sum_{j=1}^{J}(r_{tj} - W_j)x_j\Big| \le \mu,$$

or, by converting to a linear inequality system

$$-y_j \le \sum_{j=1}^{J}(r_{tj} - W_j)x_j \le y_t,\ \forall t = 1,\dots,T, \quad \frac{1}{T}\sum_{t=1}^{T} y_t \le \mu.$$

In a compact matrix form, it reads

$$-y \leq \Big(I - \frac{1}{T}E\Big)Rx \leq y, \quad \frac{1}{T}e^T y \leq \mu,$$

where E is the matrix of ones and I the identity matrix. In order to obtain a robust formulation of the portfolio selection problem, we state the linear programming problem as

$$\begin{aligned} \max \ & \sum_{j=1}^{J} W_j x_j = \frac{1}{T}e^T Rx \\ \text{subject to } & -y \leq \Big(I - \frac{1}{T}E\Big)z \leq y, \quad Rx = z, \\ & e^T x = 1, \quad \frac{1}{T}e^T y \leq \mu, \quad x, y \geq 0. \end{aligned}$$

For concreteness, consider a portfolio selection problem with $J = 4$ investments and $T = 5$ time periods. The risk aversion parameter is set as $\mu := 2$. The returns are displayed below:

time period t	reward on investment			
	1	2	3	4
1	10	20	9	11
2	12	25	11	14
3	9	17	12	12
4	11	21	11	14
5	11	19	13	16

The optimal solution is $x^* = (0,\ 0.9643,\ 0.0357,\ 0)^T$ and the corresponding optimal return is ≈ 20.08.

Suppose that the returns are not known precisely. We extend the values of r_{tj} to intervals $[0.99r_{tj}, 1.01r_{tj}]$, that is, the returns may vary independently and simultaneously within 1% tolerance. We calculate the following enclosure of the optimal solutions

$$\boldsymbol{x}^{(1)} = ([0, 0.1699],\ [0.7621, 1],\ [0, 0.181],\ [0, 0.2379])^T.$$

Even though the result is very conservative, we can conclude some interesting properties. Regardless the setting of values from intervals, we can be sure that at least 75% of the optimal portfolio is directed to the second investment. The remaining investments constitute at most 17%, 18% and 24% of the portfolio, respectively.

Now, let study 5% perturbations of all returns except for the second investment. Thus, we replace the values r_{jt} by intervals $[0.95r_{tj}, 1.05r_{tj}]$, $j = 1, 3, 4$, $t = 1, \ldots, T$. The calculated enclosure for the optimal solutions is very tight:

$$\boldsymbol{x}^{(2)} = ([0, 0.0495],\ [0.9276, 0.9712],\ [0, 0.0531],\ [0, 0.0724])^T.$$

Notice that this problem setting is not B-stable [9], that is, there is no basis being optimal for each interval realization. For instance, putting the fourth investment returns to the upper limit and the others to the lower limit, we get a solution $x^* = (0,\ 0.9492,\ 0,\ 0.0508)^T$. In this setting, it is optimal to invest in the fourth subject instead of the third one.

3 Conclusion

We proposed a method for contracting an interval enclosure of the optimal solution set. It was based on a linearization and iterative refinement. Even though it doesn't converge to the optimal bounds in general, it gives a sufficiently tight enclosure in short time. Thus it can be used as a method for solving interval linear programming itself, or a first step in more involved algorithms.

The method seems to converge quickly in spite of a very huge initial enclosure. However, to decrease the number of iterations, we will address the future research to finding a more appropriate initial enclosure. We will also carry out more numerical experiments to empirically verify the convergence speed.

Acknowledgements

The author was partially supported by the Czech Science Foundation Grant P403/12/1947.

References

[1] Beaumont, O.: Solving interval linear systems with linear programming techniques. *Linear Algebra and its Applications* **281** (1998), 293–309.

[2] Beeck, H.: Linear programming with inexact data. technical report TUM-ISU-7830, Technical University of Munich, Munich, 1978.

[3] Černý, M.: Computational aspects of regression analysis of interval data. *Proceedings of World Academy of Science, Engineering and Technology* **81** (2011), 421–428.

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

[5] Gabrel, V. and Murat, C.: Robustness and duality in linear programming. *Journal of the Operational Research Society* **61** (2010), 1288–1296.

[6] Hladík, M.: Tolerances in portfolio selection via interval linear programming. In: *CD-ROM Proceedings 26-th Int. Conf. Mathematical Methods in Economics MME08, Liberec, Czech Republic*, 2008, 185–191.

[7] Hladík, M.: Optimal value range in interval linear programming. *Fuzzy Optimization and Decision Making* **8** (2009), 283–294.

[8] Hladík, M.: How to determine basis stability in interval linear programming. Technical report KAM-DIMATIA Series (2010-973), Department of Applied Mathematics, Charles University, Prague, 2010.

[9] Hladík, M.: Interval linear programming: A survey. Technical report KAM-DIMATIA Series (2010-981), Department of Applied Mathematics, Charles University, Prague, 2010.

[10] Hladík, M.: Interval linear programming: A survey. In: *Linear Programming - New Frontiers in Theory and Applications* (Z. A. Mann, ed.), Nova Science Publishers, New York, 2012, chapter 2, 85–120.

[11] Hladík, M. and Černý, M.: Interval regression by tolerance analysis approach. *Fuzzy Sets and Systems* **193** (2012), 85–107.

[12] Jansson, C.: Rigorous lower and upper bounds in linear programming. *SIAM Journal on Optimization* **14** (2004), 914–935.

[13] Jansson, C.: A self-validating method for solving linear programming problems with interval input data. In: *Scientific computation with automatic result verification* (U. Kulisch, and H. J. Stetter, eds.), Springer, Wien, 1988, 33–45.

[14] Koníčková, J.: Sufficient condition of basis stability of an interval linear programming problem. *ZAMM, Zeitschrift fuer Angewandte Mathematik und Mechanik* **81** (2001), 677–678.

[15] Li, Y. P., Huang, G. H., Guo, P., Yang, Z. F., and Nie, S. L.: A dual-interval vertex analysis method and its application to environmental decision making under uncertainty. *European Journal of Operational Research* **200** (2010), 536–550.

[16] Liu, S.-T. and Kao, C.: Matrix games with interval data. *Computers & Industrial Engineering* **56** (2009), 1697–1700.

[17] Rohn, J.: Stability of the optimal basis of a linear program under uncertainty. *Operations Research Letters* **13** (1993), 9–12.