How to determine basis stability in interval linear programming

Milan Hladík
Charles University, Faculty of Mathematics and Physics,
Malostranské nám. 25, 118 00, Prague, Czech Republic,
e-mail: milan.hladik@matfyz.cz.

Abstract

Interval linear programming (ILP) was introduced in order to deal with linear programming problems with uncertainties that are modelled by ranges of admissible values. Basic tasks in ILP such as calculating the optimal value bounds or set of all possible solutions may be computationally very expensive. However, if some basis stability criterion holds true then the problems becomes much more easy to solve. In this paper, we propose a method for testing basis stability. Even though the method is exponential in the worst case (not surprisingly), it is fast in many cases.

Keywords: *Linear programming, linear interval systems, interval analysis, basis stability.*

1 Introduction

Many practical problems are formulated as linear programming problems. Since real-life data are often subject to uncertainties and measurement errors, we have to take it into account in linear programming methodology. Interval linear programming (ILP) was introduced to tackle these troubles. Herein, intervals represent ranges of admissible variations for particular coefficients.

There have been developed diverse methods for solving ILP problems. Some of them are based on interval arithmetic and extensions of the simplex algorithm to the case of interval data [2, 5, 6, 12], while another use a direct inspection [7, 8, 15, 25]. A different direction aims to find a satisficing solution [9, 33, 35] or a robust one [3, 13].

An important sub-class of ILP problems is that for which some basis stability condition holds true. Under the assumption of basis stability we can easily determine range of optimal values and set of all possible solutions. Without the assumption the problems becomes much more difficult. The issue of basis stability was discussed e.g. in [2, 7, 8, 15, 25].

Applications are found in diverse fields. Interval linear programming was applied in portfolio selection problem [10], network topology of transmission systems [19], air quality management [11], solid waste management planning [34] or inventory management [3].

Let us introduce some notation: I denotes the identity matrix (with convenient dimension), $\rho(A)$ the spectral radius of a matrix A and $\operatorname{diag}(v)$ the diagonal matrix with entries $v_1, \ldots, v_n$. The sign of a real r is defined as $\operatorname{sgn}(r) := 1$ if $r \geq 0$ and $\operatorname{sgn}(r) := -1$ otherwise.

2 Interval linear programming

In order to develop a sophisticated methodology for interval linear programming we have to remind some interval computing notation and results first. *An interval matrix* is defined as

$$\boldsymbol{A} = [\underline{A}, \overline{A}] = \{A \in \mathbb{R}^{m \times n};\ \underline{A} \leq A \leq \overline{A}\},$$

where $\underline{A} \leq \overline{A}$ are given matrices. By

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

we denote the center and the radius of $\boldsymbol{A}$, respectively. The set of all m-by-n interval matrices is denoted by $\mathbb{IR}^{m \times n}$. Interval vectors are defined in a similar way.

Interval arithmetic is defined in the following way [1, 14, 17]. Let $\boldsymbol{a}, \boldsymbol{b} \in$

$\mathbb{IR}$, then

$$\begin{aligned}
\boldsymbol{a} \pm \boldsymbol{b} &:= [\underline{a} \pm \underline{b}, \overline{a} \pm \overline{b}],\\
\boldsymbol{a} \cdot \boldsymbol{b} &:= [\min(\underline{a}\underline{b}, \underline{a}\overline{b}, \overline{a}\underline{b}, \overline{a}\overline{b}),\ \max(\underline{a}\underline{b}, \underline{a}\overline{b}, \overline{a}\underline{b}, \overline{a}\overline{b})],\\
\boldsymbol{a} / \boldsymbol{b} &:= \begin{cases} [\min(\underline{a}/\underline{b}, \underline{a}/\overline{b}, \overline{a}/\underline{b}, \overline{a}/\overline{b}),\ \max(\underline{a}/\underline{b}, \underline{a}/\overline{b}, \overline{a}/\underline{b}, \overline{a}/\overline{b})] & \text{if } 0 \notin \boldsymbol{b},\\ \text{undefined} & \text{otherwise.} \end{cases}
\end{aligned}$$

A basic task in interval computation is that of solving interval linear equations. Let an interval linear system

$$\boldsymbol{A}x = \boldsymbol{b} \tag{1}$$

be given, where $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$ and $\boldsymbol{b} \in \mathbb{IR}^{n}$. The solution set to a linear system is defined as a set of solutions of all scenarios of the interval data

$$\{x \in \mathbb{R}^n;\ \exists A \in \boldsymbol{A}\, \exists b \in \boldsymbol{b} : Ax = b\}.$$

A well-known description of the solution set was given by Oettli and Prager [20, 28].

Theorem 1 (Oettli and Prager, 1964)**.** *The solution set to $\boldsymbol{A}x = \boldsymbol{b}$ is described by*

$$|A^c x - b^c| \le A^\Delta |x| + b^\Delta. \tag{2}$$

Since the Oettli and Prager system is non-linear, it is cumbersome and expensive to find exact bounds of the solution set. If a non-negativity of variables is incorporated then the problem becomes easy and the characterization becomes linear [28]:

$$\underline{A}x \le \overline{b},\ -\overline{A}x \le -\underline{b},\ x \ge 0.$$

What is crucial is not non-negativity but any sign restriction of variables. Let $q \in \{\pm 1\}^n$ and consider the orthant of $\mathbb{R}^n$ defined by $\operatorname{diag}(q)\, x \ge 0$. That part of the solution set lying inside the orthant is described

$$|A^c x - b^c| \le A^\Delta \operatorname{diag}(q)\, x + b^\Delta,\ \operatorname{diag}(q)\, x \ge 0,$$

or

$$(A^c - A^\Delta \operatorname{diag}(q))x \le \overline{b}, \tag{3a}$$

$$(-A^c - A^\Delta \operatorname{diag}(q))x \le -\underline{b}, \tag{3b}$$

$$\operatorname{diag}(q)\, x \ge 0. \tag{3c}$$

The interval hull of a set is the smallest interval vector containing the set. Computing the interval hull of the solution set to (1) is NP-hard problem [31]. The interval hull $\boldsymbol{h}$ to (1) can be calculated e.g. by decomposing the space into 2^n orthants, and solving a series of linear programs as follows

$$\underline{h}_i = \min_{q\in\{\pm 1\}^n} \inf x_i \text{ subject to constraints (3)}, \qquad i = 1,\dots,n, \tag{4}$$

$$\overline{h}_i = \max_{q\in\{\pm 1\}^n} \sup x_i \text{ subject to constraints (3)}, \qquad i = 1,\dots,n. \tag{5}$$

In many practical circumstances, one doesn't need to find an exact interval hull, but any sufficiently sharp superset (called *an enclosure*) is desirable, too. Such enclosures are much faster computed, and there are plenty of methods available; see e.g. [17, 18, 23, 26] and references therein. We will employ the effective Hansen–Bliek–Rohn method [18, 24, 26].

Theorem 2 (Rohn, 1993)**.** *Let A^c be non-singular and $\rho(|(A^c)^{-1}|A^\Delta) < 1$. Denote*

$$\begin{aligned} M &:= (I - |(A^c)^{-1}|A^\Delta)^{-1}, \\ x^c &:= (A^c)^{-1}b^c, \\ x^* &:= M(|x^c| + |(A^c)^{-1}|b^\Delta). \end{aligned}$$

Then the interval vector $\boldsymbol{r}$ defined entrywise as

$$\underline{r}_i := \min\left\{-x_i^* + (x_i^c + |x_i^c|)m_{ii},\ \frac{1}{2m_{ii}-1}\left(-x_i^* + (x_i^c + |x_i^c|)m_{ii}\right)\right\},\ i = 1,\dots,n$$

$$\overline{r}_i := \max\left\{x_i^* + (x_i^c - |x_i^c|)m_{ii},\ \frac{1}{2m_{ii}-1}\left(x_i^* + (x_i^c - |x_i^c|)m_{ii}\right)\right\},\quad i = 1,\dots,n$$

forms an enclosure to the solution set.

By *an inner enclosure* of the solution set we mean an interval vector that is a subset of its interval hull. Note that an inner enclosure needn't be a subset of the solution set itself. There are few results in this issue; we will use that by Rohn [26, 30].

Theorem 3 (Rohn, 2000)**.** *Under assumption and notations from Theorem 2, define*

$$\underline{s}_i := \underline{r}_i + \left(M|(\operatorname{diag}(p^i)(A^c)^{-1}\operatorname{diag}(p^i) - |(A^c)^{-1}|)(\underline{\xi}_i A^\Delta M e_i + A^\Delta x^* + b^\Delta)|\right)_i$$

$$\overline{s}_i := \overline{r}_i - \left(M|(\operatorname{diag}(q^i)(A^c)^{-1}\operatorname{diag}(q^i) - |(A^c)^{-1}|)(\overline{\xi}_i A^\Delta M e_i + A^\Delta x^* + b^\Delta)|\right)_i,$$

where

$$\begin{aligned}
\underline{\xi}_i &:= (|\underline{h}| + \underline{h} - x^c - |x^c|)_i,\\
\overline{\xi}_i &:= (|\overline{h}| - \overline{h} + x^c - |x^c|)_i,\\
p^i_j &:= \begin{cases} \operatorname{sgn}(x^c_j) & \text{if } j \neq i,\\ -1 & \text{if } j = i,\end{cases}\\
q^i_j &:= \begin{cases} \operatorname{sgn}(x^c_j) & \text{if } j \neq i,\\ 1 & \text{if } j = i.\end{cases}
\end{aligned}$$

If $\underline{s} \leq \overline{s}$ *then* $\boldsymbol{s} = [\underline{s}, \overline{s}]$ *is an inner enclosure.*

Eventually, we introduce an interval linear programming problem. Consider a linear program

$$\min c^T x \ \text{ subject to } \ Ax = b,\ x \geq 0, \tag{6}$$

and interval entities $\boldsymbol{A} \in \mathbb{IR}^{m\times n}$, $\boldsymbol{b} \in \mathbb{IR}^m$, and $\boldsymbol{c} \in \mathbb{IR}^n$. *An interval linear programming (ILP)* is a family of linear programs (6), where $A \in \boldsymbol{A}$, $b \in \boldsymbol{b}$, $c \in \boldsymbol{c}$. *A scenario* is a concrete realization of interval values, that is, any linear program (6) with $A \in \boldsymbol{A}$, $b \in \boldsymbol{b}$, $c \in \boldsymbol{c}$.

3 Basis stability

By a *basis* we mean an index set $B \subseteq \{1, \ldots, n\}$ such that A_B is nonsingular, where A_B denotes the restriction of A to the columns indexed by B. Analogously, $N := \{1, \ldots, n\} \setminus B$ stands for non-basic variables, and A_N denotes the restriction to non-basic indices. A basis B is *feasible* if $A_B^{-1} b \geq 0$ holds, and B is called *optimal* as long as it is feasible and $c_N^T - c_B^T A_B^{-1} A_N \geq 0^T$ comes true.

Definition 1. Let a basis B be given. The ILP problem is called *B-stable* if B is an optimal basis for each scenario of interval values. ILP is called *[unique] non-degenerate B-stable* if each scenario has a [unique] optimal basic solution with the basis B.

In general, B-stability has not been investigated thoroughly yet. Non-degenerate B-stability was studied e.g. in [25, 7]. The following reduction to 2^{2m} linear programs is due to Rohn [25].

Theorem 4 (Rohn, 1993)**.** *Let a basis B be given. ILP is [unique] non-degenerate B-stable iff it is true for the following finite set of scenarios*

$$\begin{aligned} &\min\ (c^c + \operatorname{diag}(q)\, c^\Delta)^T x \\ &\textit{subject to}\ \ (A^c - \operatorname{diag}(p)\, A^\Delta \operatorname{diag}(q))x = b^c + \operatorname{diag}(p)\, b^\Delta, x \geq 0, \end{aligned}$$

where $p \in \{\pm 1\}^m$ and $q \in \{q \in \mathbb{R}^n;\ |q_j| = 1\ \forall j \in B,\ q_j = 1\ \forall j \notin B\}$.

The subsequent sufficient condition is due to Koníčková [7]. Since it requires an interval hull calculation, the complexity is still exponential. Note that the left-hand side of (7) is an upper limit of the interval matrix product calculation.

Theorem 5 (Koníčková, 2001)**.** *Let B be a basis, $\boldsymbol{x}_B$ an interval hull of the solution set to $\boldsymbol{A}_B x_B = \boldsymbol{b}$, and $\boldsymbol{y}$ to $\boldsymbol{A}_B^T y = \boldsymbol{c}_B$. Suppose that $\boldsymbol{A}_B$ is regular, and $\underline{x}_B > 0$. If the inequality*

$$\overline{(\boldsymbol{A}_N^T)\boldsymbol{y}} \leq \underline{c}_N \tag{7}$$

holds then ILP is non-degenerate B-stable with a basis B. If (7) holds strictly then it is unique non-degenerate B-stable.

B-stability of ILP is very important because it enables to describe the set of all possible optimal solution [2, 7, 15] and to calculate the optimal value range. Under the assumption of unique B-stability, the set of all optimal solutions is equal to the solution set of the interval system $\boldsymbol{A}_B x_B = \boldsymbol{b}$, $x_B \geq 0$, $x_N = 0$. By Rohn [28], the set is formed by a convex polyhedral set described by

$$\underline{A}_B x_B \leq \overline{b},\ -\overline{A}_B x_B \leq -\underline{b},\ x_B \geq 0,\ x_N = 0.$$

When the ILP problem is B-stable, but not unique B-stable, then each scenario of ILP has at least one optimal solution in this set, and, conversely, each solution of the set is an optimal solution of some scenario.

Optimal value bounds are defined as [4, 16, 27]

$$\begin{aligned} \underline{f} &:= \inf f(A, b, c)\ \text{ subject to }\ A \in \boldsymbol{A},\ b \in \boldsymbol{b},\ c \in \boldsymbol{c}, \\ \overline{f} &:= \sup f(A, b, c)\ \text{ subject to }\ A \in \boldsymbol{A},\ b \in \boldsymbol{b},\ c \in \boldsymbol{c}. \end{aligned}$$

The lower bound is efficiently computable while the upper bound is NP-hard [27]. Nevertheless, under the assumption of B-stability we obtain the following efficient formulae

$$\begin{aligned} \underline{f} &= \min \underline{c}_B^T x_B \ \text{ subject to } \ \underline{A}_B x_B \leq \overline{b},\ -\overline{A}_B x_B \leq -\underline{b},\ x_B \geq 0,\\ \overline{f} &= \max \overline{c}_B^T x_B \ \text{ subject to } \ \underline{A}_B x_B \leq \overline{b},\ -\overline{A}_B x_B \leq -\underline{b},\ x_B \geq 0. \end{aligned}$$

4 Our method

Let a basis B be given. It can be computed by an interval version of the simplex method or estimated by solving a scenario with the midpoint values, for instance.

In a real-valued linear programming, a basis B is optimal basis if and only if three conditions holds:

1. (regularity) A_B is non-singular;
2. (feasibility) $A_B^{-1}b \geq 0$;
3. (optimality) $c_N^T - c_B^T A_B^{-1} A_N \geq 0^T$.

Extension to interval data leads to the following characterization of B-stability. The basis B is optimal for each scenario if and only if conditions 1. to 3. hold for each $A \in \boldsymbol{A}$, $b \in \boldsymbol{b}$, and $c \in \boldsymbol{c}$. In the sequel we will discuss the three conditions in detail.

4.1 Regularity

An interval matrix $\boldsymbol{M} \in \mathbb{IR}^{n\times n}$ is *regular* if every $M \in \boldsymbol{M}$ is non-singular. It was proved by Poljak and Rohn [21] that checking regularity is NP-hard problem, so the first condition cannot be answered efficiently. However, there is a plenty of diverse methods for testing regularity; see e.g. a review paper by Rohn [29] or [26]. Moreover, there are several sufficient conditions that can be employed as well [22, 26]. For instance, a broadly used one is that if the spectral radius of $|(M^c)^{-1}|M^\Delta$ is less than 1 then $\boldsymbol{M}$ is regular.

Theorem 6. *If $\rho\left(|(A^c)_B^{-1}|A_B^\Delta\right) < 1$ then $\boldsymbol{A}_B$ is regular.*

A useful necessary condition is the following one [22, 26].

Theorem 7. *If* $\max_{i=1,\dots,n}\left(|(A^c)_B^{-1}|A_B^\Delta\right)_{ii} \geq 1$ *holds then* $\boldsymbol{A}_B$ *is not regular.*

4.2 Feasibility

Turning to the second point, the inequality $A_B^{-1}b \geq 0$ holds for every $A \in \boldsymbol{A}$ and $b \in \boldsymbol{b}$ if and only if the solution set to the interval system $\boldsymbol{A}_B x_B = \boldsymbol{b}$ lies in the non-negative orthant. The simple but exponential method is to compute the exact interval hull of the solution set and check for non-negativity. Another way is to utilize some solver for interval equations to get an enclosure of the solution set, and again to check its non-negativity. This leads to a fast sufficient condition. As a necessary condition it can serve any inner enclosure of the solution set; if it is not non-negative then the interval hull cannot be, too. In our algorithm we utilize Theorem 2 for enclosure computation and Theorem 3 for inner enclosure computation.

4.3 Optimality

In the third point, the condition is equivalent to

$$c_N^T \geq yA_N,\ A_B^T y = c_B. \tag{8}$$

The interval counterpart takes the form

$$\boldsymbol{A}_N^T y \leq \boldsymbol{c}_N,\ \boldsymbol{A}_B^T y = \boldsymbol{c}_B.$$

There are no dependencies in this system (A_B and A_N are disjunctive), and so it is a standard interval system. Few is known about its feasibility as it consists of mixed equations and inequalities. However, we will exploit the regularity of A_B to characterize strong feasibility.

First, we derive a simple sufficient condition. Let $\boldsymbol{y}$ be an enclosure to the solution set of $\boldsymbol{A}_B^T y = \boldsymbol{c}_B$. If

$$\overline{(\boldsymbol{A}_N^T)\boldsymbol{y}} \leq \underline{c}_N \tag{9}$$

then in each scenario the solution to the equation system solves also the whole system (8) and thus the strong feasibility is valid. Observe that we came across the same relation as in (7), but with weaker assumptions. Moreover, we are able to derive a complete characterization instead of a sufficient condition only.

Similarly we will proceed to derive a sufficient and necessary characterization to the third condition. We formalize the property that for each scenario the solution to the equation system solves the whole system (8) as well. By Theorem 1, the solution set of $\boldsymbol{A}_B^T y = \boldsymbol{c}_B$ is described by

$$|(A_B^c)^T y - c_B^c| \leq (A_B^\Delta)^T |y| + c_B^\Delta \tag{10}$$

According to Rohn [28, 32], a vector y solves each scenario of $\boldsymbol{A}_N^T y \leq \boldsymbol{c}_N$ if and only if it solves

$$(A_N^c)^T y - c_N^c \leq -(A_N^\Delta)^T |y| - c_N^\Delta. \tag{11}$$

Both the sets are non-convex polyhedral sets, however, they are convex when restricted to an orthant. Let $q \in \{\pm 1\}^m$ and consider the orthant $\operatorname{diag}(q)\, y \geq 0$. Restriction of (10) to the orthant reads

$$\big((A_B^c)^T - (A_B^\Delta)^T \operatorname{diag}(q)\big)\, y \leq \overline{c}_B, \tag{12a}$$

$$-\big((A_B^c)^T + (A_B^\Delta)^T \operatorname{diag}(q)\big)\, y \leq -\underline{c}_B, \tag{12b}$$

$$\operatorname{diag}(q)\, y \geq 0 \tag{12c}$$

while the restriction of (11) reads

$$\big((A_N^c)^T + (A_N^\Delta)^T \operatorname{diag}(q)\big)\, y \leq \underline{c}_N,\ \operatorname{diag}(q)\, y \geq 0. \tag{13}$$

Thus the proposed method is based on partitioning of the space $\mathbb{R}^m$ into 2^m orthants and testing whether (12) lies in (13) for each $q \in \{\pm 1\}^m$. Polyhedra inclusion testing can be performed by solving a linear number of linear programs, namely $n - m$.

4.4 Summary

Algorithm 1 summarizes the method described above. Even though it is designed for B-stability, it can be easily adapted for [unique] non-degenerate B-stability, too. A basis B is optimal and corresponds to a non-degenerate solution if and only if there is a strict inequality in the feasibility condition, that is, $A_B^{-1} b > 0$. Thus Algorithm 1 will work for non-degenerate B-stability when we put strict inequality in steps 16, 20 and 24.

For a unique basis stability no such a simple characterization is known. A simple but strong sufficient condition is that there is a strict inequality in the optimality test (8). Replacing by the strict inequality in step 33 and strict inclusion in step 37 we adapt the algorithm to unique B-stability. However, we must also replace step 38 by "no decision" since we cannot decide in this case.

The proposed algorithm runs quickly in many cases, but it is exponential in the worst case. Each of the three main steps requires solving up to 2^m linear programs. Thus, the overall complexity is $\mathcal{O}(2^m poly(m,n))$, where $poly(m,n)$ is the time for a linear program; it is polynomial as long as we employ a suitable solver. The complexity is still much better than that for the Rohn's method (Theorem 4), and it is comparable with the method by Koníčková (Theorem 5), which gives only a sufficient condition.

Provided $P \neq NP$ there cannot exist an efficient algorithm for testing basis stability since the problem is NP-hard. Below, we give a proof for B-stability. [Unique] non-degenerate B-stability seems to be NP-hard, too, but the proof remains open.

Proposition 1. *Checking B-stability for a given basis B in NP-hard.*

Proof. We construct a reduction from the problem of testing regularity of an interval matrix, which is known to be NP-hard [21]. Let $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$. We claim that $\boldsymbol{A}$ is regular if and only if the ILP problem

$$\min 0^T x + 0^T y \ \text{ subject to } \ \boldsymbol{A}x + Iy = 0,\ x, y \geq 0$$

is B-stable with $B = (1, \dots, n)$. Let $A \in \boldsymbol{A}$. If A is non-singular then B is optimal basis since $x = y = 0$ is basic feasible solution, and every feasible point is optimal. Conversely, if A is singular then B cannot be optimal basis. $\square$

5 Example

Example 1. Consider an ILP problem

$$\min \boldsymbol{c}^T x \ \text{ subject to } \ \boldsymbol{A}x = \boldsymbol{b},\ x \geq 0$$

with data

$$\boldsymbol{A} = \begin{pmatrix} -[3,4] & [7,8] & [5,6] \\ [6,7] & -[7,8] & [1,2] \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} [7,8] \\ [5,6] \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} [3,4] \\ [5,6] \\ [1,2] \end{pmatrix}.$$

A candidate basis for B-stability is $B = (1,3)$ since it is optimal for the scenario taking the center values. Let us proceed along the depicted algorithm and check for validity of the three points.

1. According to Theorem 6, we calculate the spectral radius 0.2073, so the interval matrix $\boldsymbol{A}_B$ is regular.

2. By the Hansen–Bliek–Rohn method (Theorem 2) we compute an enclosure to the solution set of $\boldsymbol{A}_B x_B = \boldsymbol{b}$ to be $\boldsymbol{x}_B = ([0.1867, 0.7997], [1.2912, 2.1389])^T$. It is non-negative, so the second point is valid.

3. By the Hansen–Bliek–Rohn method we compute an enclosure to the solution set of $\boldsymbol{A}_B^T y = \boldsymbol{c}_B$ to be $\boldsymbol{y} = ([-0.0734, 0.3199], [0.4124, 0.8340])^T$. Now, the relation (9) reads

$$\overline{(\boldsymbol{A}_N^T)\boldsymbol{y}} = -0.1171 \leq \underline{c}_N = 5.$$

Thus the sufficient condition is valid.

All three conditions are satisfied and hence the problem is B-stable. Changing the interval coefficient $\boldsymbol{b}_1$ to $\boldsymbol{b}_1 := [7, 11]$ the problem is still quickly verified to be B-stable. When we change $\boldsymbol{b}_1$ to $\boldsymbol{b}_1 := [7, 12]$ then in the second point we obtain the enclosure $\boldsymbol{x}_B = ([-0.0034, 0.8680], [1.2912, 2.8706])^T$, which is not non-negative. Thus we have to calculate the exact interval hull $([0.0232, 0.7436], [1.3333, 2.8236])^T$. Since it is non-negative now, we can confirm B-stability again. However, changing $\boldsymbol{b}_1$ to $\boldsymbol{b}_1 := [7, 13]$ the interval hull reads $([-0.0278, 0.7436], [1.3333, 3.0001])^T$, and the problem will no more be B-stable. Unfortunately, we have to calculate the exact interval hull as the necessary condition based on inner enclosure (step 20 of Algorithm 1) is not successful here.

Let us inspect also effects of changes in the objective function coefficients. The problem is basis stable even when replacing $\boldsymbol{c}_3$ by $\boldsymbol{c}_3 := [1, 5]$, but for $\boldsymbol{c}_3 := [1, 6]$ the sufficient condition (9) is not fulfilled. Thus we decompose the problem into testing $2^m = 4$ inclusion properties of a pair of convex polyhedral sets. It turns out that all the inclusions are fulfilled, whence the B-stability follows. Eventually, when we change $\boldsymbol{c}_3$ to $\boldsymbol{c}_3 := [1, 10]$ then it fails to be basis stable.

The example illustrates that the sufficient conditions for basis stability are quite strong, but when the data perturbations are large enough then we must call the exact but costly algorithm. □

6 Conclusion

We proposed a new algorithm for testing basis stability of an interval linear programming problem. Even though the complexity is (and due to NP-hardness probably must be) exponential, it runs quickly in many cases and outperforms the known methods. The algorithm works not only for basis stability but non-degenerate basis stability, too.

There are some open problems remaining. A necessary condition for optimality testing in Section 4.3 would be desirable as it may accelerate the decision process. A complete method for testing unique [non-degenerate] basis stability is of interest, too.

References

[1] G. Alefeld and J. Herzberger. *Introduction to interval computations.* Academic Press, London, 1983.

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

[3] V. Gabrel, C. Murat, and N. Remli. Linear programming with interval right hand sides. *Int. Trans. Oper. Res.*, 17(3):397–408, 2010.

[4] M. Hladík. Optimal value range in interval linear programming. *Fuzzy Optim. Decis. Mak.*, 8(3):283–294, 2009.

[5] C. Jansson. A self-validating method for solving linear programming problems with interval input data. In U. Kulisch and H. J. Stetter, editors, *Scientific computation with automatic result verification*, Computing Suppl. 6, pages 33–45, Wien, 1988. Springer.

[6] C. Jansson and S. M. Rump. Rigorous solution of linear programming problems with uncertain data. *Z. Oper. Res.*, 35(2):87–111, 1991.

[7] J. Koníčková. Sufficient condition of basis stability of an interval linear programming problem. *ZAMM, Z. Angew. Math. Mech.*, 81(Suppl. 3):677–678, 2001.

[8] R. Krawczyk. Fehlerabschätzung bei linearer optimierung. In *Interval Mathematics*, LNCS 29, pages 215–222, Berlin, 1975. Springer.

[9] D. Kuchta. A modification of a solution concept of the linear programming problem with interval coefficients in the constraints. *CEJOR, Cent. Eur. J. Oper. Res.*, 16(3):307–316, 2008.

[10] K. K. Lai, S. Y. Wang, J. P. Xu, S. S. Zhu, and Y. Fang. A class of linear interval programming problems and its application to portfolio selection. *IEEE Trans. Fuzzy Syst.*, 10(6):698–704, 2002.

[11] Y. P. Li, G. H. Huang, P. Guo, Z. F. Yang, and S. L. Nie. A dual-interval vertex analysis method and its application to environmental decision making under uncertainty. *Eur. J. Oper. Res.*, 200(2):536–550, 2010.

[12] B. Machost. Numerische Behandlung des Simplexverfahrens mit intervallanalytischen Methoden. Technical Report 30, Berichte der Gesellschaft für Mathematik und Datenverarbeitung, 54 pages, Bonn, 1970.

[13] R. Montemanni and L. M. Gambardella. An exact algorithm for the robust shortest path problem with interval data. *Comput. Oper. Res.*, 31(10):1667–1680, 2004.

[14] R. E. Moore. *Interval analysis*. Englewood Cliffs, N. J., 1966.

[15] F. Mráz. On infimum of optimal objective function values in interval linear programming. Technical report KAM Series (96-337), Department of Applied Mathematics, Charles University, Prague, 1996.

[16] F. Mráz. Calculating the exact bounds of optimal values in LP with interval coefficients. *Ann. Oper. Res.*, 81:51–62, 1998.

[17] A. Neumaier. *Interval methods for systems of equations*. Cambridge University Press, Cambridge, 1990.

[18] S. Ning, Kearfott, and R. Baker. A comparison of some methods for solving linear interval equations. *SIAM J. Numer. Anal.*, 34(4):1289–1305, 1997.

[19] A. S. Noghabi, H. R. Mashhadi, and J. Sadeh. Optimal coordination of directional overcurrent relays considering different network topologies using interval linear programming. *IEEE Trans. Power Deliv.*, 25(3):1348–1354, 2010.

[20] W. Oettli and W. Prager. Compatibility of approximate solution of linear equations with given error bounds for coefficients and right-hand sides. *Numer. Math.*, 6:405–409, 1964.

[21] S. Poljak and J. Rohn. Checking robust nonsingularity is NP-hard. *Math. Control Signals Syst.*, 6(1):1–9, 1993.

[22] G. Rex and J. Rohn. Sufficient conditions for regularity and singularity of interval matrices. *SIAM J. Matrix Anal. Appl.*, 20(2):437–445, 1998.

[23] J. Rohn. Systems of linear interval equations. *Linear Algebra Appl.*, 126(C):39–78, 1989.

[24] J. Rohn. Cheap and tight bounds: The recent result by E. Hansen can be made more efficient. *Interval Comput.*, 1993(4):13–21, 1993.

[25] J. Rohn. Stability of the optimal basis of a linear program under uncertainty. *Oper. Res. Lett.*, 13(1):9–12, 1993.

[26] J. Rohn. A handbook of results on interval linear problems. `http://www.cs.cas.cz/rohn/handbook`, 2005.

[27] J. Rohn. Interval linear programming. In M. Fiedler et al., editor, *Linear optimization problems with inexact data*, chapter 3, pages 79–100. Springer, New York, 2006.

[28] J. Rohn. Solvability of systems of interval linear equations and inequalities. In M. Fiedler et al., editor, *Linear optimization problems with inexact data*, chapter 2, pages 35–77. Springer, New York, 2006.

[29] J. Rohn. Forty necessary and sufficient conditions for regularity of interval matrices: A survey. *Electron. J. Linear Algebra*, 18:500–512, 2009.

[30] J. Rohn. An improvement of the Bauer-Skeel bounds. Technical Report V-1065, Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague, 2010.

[31] J. Rohn and V. Kreinovich. Computing exact componentwise bounds on solutions of linear systems with interval data is NP-hard. *SIAM J. Matrix Anal. Appl.*, 16(2):415–420, 1995.

[32] J. Rohn and J. Kreslová. Linear interval inequalities. *Linear Multilinear Algebra*, 38(1-2):79–82, 1994.

[33] A. Sengupta and T. K. Pal. *Fuzzy Preference Ordering of Interval Numbers in Decision Problems*, volume 238 of *Studies in Fuzziness and Soft Computing*. Springer, Berlin, 2009.

[34] Y. Sun, Y. Li, and G. Huang. Dual-interval linear programming model and its application to solid waste management planning. *Environ. Eng. Sci.*, 27(6):451–468, 2010.

[35] F. Zhou, H. C. Guo, G. X. Chen, and G. H. Huang. The interval linear programming: A revisit. *J. Environ. Inform.*, 11(1):1–10, 2008.

Algorithm 1 (B-stability)

1: {regularity}
2: **if** $\rho\left(|(A^c)_B^{-1}|A_B^\Delta\right) < 1$ **then**
3: goto step 14;
4: **else if** $\max_{i=1,\dots,n}\left(|(A^c)_B^{-1}|A_B^\Delta\right)_{ii} \geq 1$ **then**
5: **return** "not B-stable";
6: **else**
7: check regularity of $\boldsymbol{A}_B$ e.g. by the method from [26];
8: **if** $\boldsymbol{A}_B$ is regular **then**
9: goto step 14;
10: **else**
11: **return** "not B-stable";
12: **end if**
13: **end if**
14: {feasibility}
15: compute the enclosure $\boldsymbol{r}$ to $\boldsymbol{A}_B x_B = \boldsymbol{b}$ by Theorem 2;
16: **if** $\underline{r} \geq 0$ **then**
17: goto step 31;
18: **else**
19: compute the inner enclosure $\boldsymbol{s}$ to $\boldsymbol{A}_B x_B = \boldsymbol{b}$ by Theorem 3;
20: **if not** $\underline{s} \geq 0$ **then**
21: **return** "not B-stable";
22: **else**
23: compute the interval hull $\boldsymbol{h}$ to $\boldsymbol{A}_B x_B = \boldsymbol{b}$ by (4)–(5);
24: **if** $\underline{h} \geq 0$ **then**
25: goto step 31;
26: **else**
27: **return** "not B-stable";
28: **end if**
29: **end if**
30: **end if**
31: {optimality}
32: compute the enclosure $\boldsymbol{r}$ to $\boldsymbol{A}_B^T y = \boldsymbol{c}_B$ by Theorem 2;
33: **if** $\overline{(\boldsymbol{A}_N^T)\boldsymbol{y}} \leq \underline{c}_N$ **then**
34: **return** "B-stable";
35: **else**
36: **for** $q \in \{\pm 1\}^m$ **do**
37: **if** (12) does not lie inside (13) **then**
38: **return** "not B-stable";
39: **end if**
40: **end for**
41: **return** "B-stable";
42: **end if**