Journal of Universal Computer Science, vol. 4, no. 2 (1998), 164-170
submitted: 25/9/97, accepted: 1/11/97, appeared: 28/2/98 © Springer Pub. Co.

Pascal Subroutines for Solving Some problems in Interval LP

František Mráz

Martin Kursch, Daniel Panuska [1]
(Dept. of Math., University of South Bohemia, Jeronymova 10, 37115 C. Budejovice, Czech republic
mraz@pf.jcu.cz)
(University of South Bohemia, C. Budejovice)

Abstract: This paper deals with the Pascal subroutines for solving certain problems in the interval linear programming, especially with calculating the exact range, i.e. the supremum and the infimum of optimal objective function values of a family of LP problems in which all coefficients in constraints vary in given intervals. A theoretical background of the algorithms and a description of the package is included. An application of algorithms regarding a set of feasible coefficients and the solvability set is described in this paper and numerical experiences are also mentioned.

Key words: linear programming problem, inexact data, interval coefficients, Pascal subroutines.

1 Introduction

Let us consider the following linear programming (LP) problem in the standard form

$$S(A, b): \qquad \sup\{c^T x : Ax = b, x \geq 0\}. \tag{1}$$

As a case of inexact input data, let us assume that the data may vary independently in given intervals. Therefore, we shall consider an interval m by n matrix $[A]$ and an interval $m-$vector $[b]$ given by the bounds $\underline{A}$, $\overline{A}$, $\underline{b}$ and $\overline{b}$ satisfying $\underline{b} < \overline{b}$.

By an *interval linear programming* (ILP) problem we mean the family of LP problems $S(A, b)$, where $A \in [A]$, $b \in [b]$ and $c \in \mathrm{R}^n$ is a given vector.

For the fixed $A \in [A]$ and $b \in [b]$, the problem $S(A, b)$ is called a *subproblem* of an ILP problem with the set of feasible solutions denoted by $X(A, b)$. Let $H = \{h \in \mathrm{R}^m : |h_i| = 1, i = 1, ..., m\}$. By using S_h, $h \in H$ let us introduce the so-called *extremal subproblem*, of which the i-th constraint has the form $(\overline{A}x)_i = \underline{b}_i$ if $h_i = 1$ and $(\underline{A}x)_i = \overline{b}_i$ if $h_i = -1, i = 1, 2, ..., m$. The corresponding set of feasible solutions will be denoted by X_h.

Let $f(A, b)$ denote the optimal value of a subproblem $S(A, b)$. The function f defined in this way is called the *solution function* of an ILP problem. The package is devoted to calculating the supremum $\overline{f}$ and the infimum $\underline{f}$ of the solution function subject to given intervals, i.e., calculating the exact upper and lower bounds of optimal values for all subproblems of a given ILP problem. For such a

[1] This research was supported by the Grant Agency of the Czech Republic under the grant No. 201/95/1484.

problem it is redundant to consider an interval vector $[c] = \{c \in \mathrm{R}^n : \underline{c} \le c \le \overline{c}\}$ in the objective function because it holds $\underline{c}^T x \le c^T x \le \overline{c}^T x$ for all $x \ge 0$ and $c \in [c]$.

The problem was solved either by using interval arithmetic, e.g., by Beeck [1], Jansson [3], [4] or without interval analysis by Rohn in [10], [11]. The mentioned results were received under a restricting assumption, e.g., a basis stable case (see Section 2) was often considered. For our Pascal package, the approach without interval arithmetic is used as well. The theoretical background and a description of the algorithms are given in full detail in [5], [6] and [7]. There is a brief summary in Section 2.

By the *set of feasible solutions* we denote the set

$$F = \{(A,b) \in [A] \times [b] : X(A,b) \neq \emptyset\}. \tag{2}$$

The *solvability set* S is defined as the set containing all values A, b, c belonging to the given intervals so that problem (1) has a finite optimum.

An application regarding the sets F and S is mentioned in Section 2. A description of the Pascal package including certain numerical experiences is given in Section 3.

2 Theoretical Background, Algorithms and Applications

Let us denote by X the set of all feasible solutions for a given ILP problem, i.e.,

$$X = \cup\{X(A,b) : A \in [A], b \in [b]\}. \tag{3}$$

It is known that the set X is a convex polytope which follows immediately from the Oettli-Prager theorem [8]. Moreover, the values $\overline{f}$ and $\underline{f}$ are reached in some vertices of X. An important class of ILP problems is introduced by the following definition: The set X is called *regular* if each of its vertices is a vertex of some $X_h, h \in H$. The calculation of the exact upper bound $\overline{f}$ is based on the following assertion:

Theorem 1. *Let x be a vertex of the regular set X. Then $c^T x = \overline{f}$ if and only if x is an optimal solution of an extremal subproblem S_h, in which the dual optimal $y = (y_i)$ satisfies the following:*

$$y_i h_i \le 0 \quad i = 1, ..., m. \tag{4}$$

Proof was given in [5].

Theorem 1 implies a possibility to solve a sequence of extremal subproblems S_h until the optimality criterion (4) is satisfied.

Algorithm SUP

0. Choose a $h \in H$.

1. Compute the optimal solution x^h of the subproblem S_h and a dual optimal solution y. If S_h is unbounded, then $\overline{f} := \infty$ and STOP.

2. If $y_i h_i \le 0, i = 1, ..., m$, then set $\overline{f} := c^T x^h$ and STOP.

3. Specify the index s with $y_s h_s = \max\{y_i h_i : y_i h_i > 0\}$ and proceed to the new subproblem, of which the constraints are changed in the s-th row.

4. Go to step 1.

The continuation to a new subproblem in step 3 can be done efficiently by adding the opposite s-th extremal equation to the constraints and by performing an usual step of the simplex method. All details are described in [5].

If the set X is regular, then the algorithm SUP operates with m rows table except for changing the s-th row in step 3, where one row is added. If the set X is not regular, then the value $\overline{f}$ can be achieved at some vertex of X which does not belong to any $X_h, h \in H$. In this case we must consider not only the extremal, but also the so-called t-subproblems. Then, the number of rows in a table may change between m and $2m$ during a calculation. It was proved in [5], however, that the optimality criterion generalized for the t-subproblems has the form (4). Moreover, for an arbitrary ILP problem, the algorithm SUP terminates in a finite number of steps in calculating the exact upper bound of optimal values $\overline{f}$ or with a conclusion that the set of all feasible solutions X is empty. We should emphasize an important feature of the algorithm SUP: If the set X is regular then the SUP algorithm works in a simplified way with m rows table without verifying the regularity of X.

Calculating the exact lower bound $\underline{f}$ of optimal values is more difficult as it is an NP-hard problem, see Rohn [12]. First, we introduce new definitions. A given ILP problem is called *strongly solvable* if each subproblem $S(A, b)$ has a finite optimum. An ILP problem will be called *basis stable with basis B* if each subproblem $S(A, b)$ has a unique nondegenerate optimal basic solution with the basis B.

Theorem 2. *Let an ILP problem be strongly solvable. Then there is an $h^* \in H$ such that $\underline{f} = f(A_{h^*}, b_{h^*}) = \min\{f(A_h, b_h) : h \in H\}$.*

Theorem 2 leads to a possibility of solving 2^m extremal subproblems to calculate the exact lower bound $\underline{f}$. For a small value of m it can be calculated efficiently by using the algorithm SUP as it proceeds easily and efficiently from an optimal solution of an extremal subproblem to a feasible (sometimes even optimal) solution of another extremal subproblem. The problem was solved by the algorithm with a systematic passage through all extremal subproblems: henceforth, the algorithm is denoted by AS. However, depending on a hardware, the value m should not be greater than 20 in order to finish a computation within a reasonable time limit.

There is another alternative to solve the problem in question. It is based on the following assertion which gives, however, only a necessary condition for calculating the exact lower bound of optimal values $\underline{f}$.

Theorem 3. *Let an ILP problem be strongly solvable. If x is an optimal solution of an extremal subproblem S_h with $c^T x = \underline{f}$, then a dual optimal solution $y = (y_i)$ satisfies the condition*

$$y_i h_i \geq 0, \quad i = 1, ..., m. \tag{5}$$

As the condition (5) differs from the condition (4) in opposite inequalities only, Theorem 3 implies a modified algorithm with obvious changes of the algorithm SUP in steps 2 and 3. Because we have a necessary condition in Theorem 3 only, there is a problem of the termination of the procedure and of a possible conclusion if the criterion (5) in step 2 is satisfied. Finiteness of the modified algorithm in reaching the exact lower bound $\underline{f}$ is ensured under the assumption of basic stability.

If an ILP problem is basis stable, then a procedure utilizing step 3 of the algorithm yields immediately an optimal solution of a new subproblem with a smaller optimal value. It should be emphasized that the algorithm works in this way without the knowledge of basic stability. Therefore, after finding an optimal solution of the initial subproblem, the solution function value decreases monotonically by passing through the vertices of the set X if condition (5) is not satisfied. Thus, the value $\overline{f}$ is reached in a finite number of steps. The problem, however, consists in the fact that the basic stability assumption is difficult to verify except for some sufficient conditions.

Neverthless, there is still a result regarding the termination of the above procedure in a general case. It was shown in [7] that the condition (5) is sufficient at least for a local lower bound of optimal values. Because of this result, the modified algorithm will be denoted by LM (i. e. Local Minimizer of the solution function).

The above algorithms enable us not only to calculate the exact lower and upper bound of optimal values but also to obtain applications. The first one is connected with the problem of whether or not each LP problem $S(A, b), A \in [A], b \in [b]$ has a feasible solution, i.e., whether or not the set of feasible solutions $F = [A] \times [b]$. The problem can also be formulated in the following way: Is the set $X(A, b)$ of nonnegative solutions of a system of linear equations $Ax = b$ nonempty for each $A \in [A], b \in [b]$? The following Theorem was proved by Rohn in [9]:

Theorem 4. *$X(A, b)$ is not empty for all $(A, b) \in [A] \times [b] \iff$ all extremal subproblems $S_h, h \in H$ have nonempty sets X_h.*

If we use the algorithm AS, then a verification of a feasibility of all extremal subproblems need not solve 2^m extremal systems, but to perform 2^m steps of the Simplex method.

The second application regards the solvability set of a given ILP problem, namely, a decision whether all LP problems

$$S(A, b, c): \qquad \sup\{c^T x : Ax = b, x \geq 0\} \tag{6}$$

where $A \in [A]$, $b \in [b]$ and $c \in [c]$ have a finite optimum. Let us consider an ILP problem of a special form with input intervals given by the central values A_0, b_0, c_0 and by relative error ϵ, i.e. $[A] = A_0 \pm \epsilon|A_0|$, $[b] = b_0 \pm \epsilon|b_0|$, $[c] = c_0 \pm \epsilon|c_0|$. Such an ILP will be indicated by ILP(ϵ) with the set of all feasible solutions $X(\epsilon)$.

Theorem 5. *If $\epsilon_1 < \epsilon_2$ then $X(\epsilon_1) \subset X(\epsilon_2)$.*

Proof follows easily from the Oettli-Prager theorem [8].

Suppose that the subproblem $S(A_0, b_0, \overline{c})$ has a finite optimum. Because of Theorem 5 the following procedure can be performed: Let us start by using the above SUP algorithm for calculating the exact upper bound of optimal values

$\overline{f}(\epsilon)$ for an ILP(ϵ) problem with an initial fixed value ϵ. If $\overline{f}(\epsilon)_0$ is finite then repeat using the SUP algorithm with the value $2\epsilon_0$, otherwise with the value $\epsilon_0/2$. After a finite number of steps, a modified procedure of the bisection of an interval for ϵ values results in calculating values ϵ_1, ϵ_2 with a required accuracy such that all ILP(ϵ) problems with $\epsilon \in [0, \epsilon_1]$ have a finite upper bound of optimal values $\overline{f}(\epsilon)$ and for each $\epsilon > \epsilon_2$ there is an unbounded subproblem of an ILP(ϵ) problem. Together with the previous application we can find a value $\epsilon_0 \leq \epsilon_1$ such that for any $\epsilon \in [0, \epsilon_0]$ all subproblems of a corresponding ILP(ϵ) problem have a finite optimum.

3 Subroutines and Numerical Experience

The ILP package in Pascal can be operated very easily by means of a basic menu offering the following options: File, Edit, Work, Export and Help.

The File option gives the following possibilities:

- to Open any file contained in the directory ILP (e.g. to observe results of calculations in output files which are contained in directories OutputAS, OutputLM and OutputSU)

- to Set up paths with a possibility for changing the adjusted paths to input and output files

- to Print a current file.

The Edit option helps in editing a file which has been opened by the File option.

The Work option enables us to run three basic subroutines denoted by SUP, LM and AS in correspondence to the above algorithms. The subroutines are based on the codes of Bunday and Garside [2] and they were tested on PCs. It is possible to use both the MS DOS and MS Windows operation systems. All new procedures are described in a manual in details which enable any user to modify subroutines, especially in procedures connected with a form of input and output files. By running a subroutine, a user is asked for names of input and output files. There are two types of input data which must be prepared in the input file.

(i) The input data of the form

$m \quad n \quad \underline{A} \quad \overline{A} \quad \underline{b} \quad \overline{b} \quad c$

if the interval matrix $[A]$ and interval vector $[b]$ are given by their bounds $\underline{A}$, $\overline{A}$, $\underline{b}$ and $\overline{b}$.

(ii) The input data of the form

$m \quad n \quad A_0 \quad b_0 \quad \epsilon \quad c$

if the interval matrix $[A]$ and interval vector $[b]$ are given by their central values A_0 and b_0 with $[A] = A_0 \pm \epsilon|A_0|$, $[b] = b_0 \pm \epsilon|b_0|$.

An output file contains final results with input data and final values of an optimal solution. Subroutines LM and AS contain a warning label if the set of feasible solutions is not regular.

By running subroutines, the user has an opportunity to choose the initial extremal subproblem. The menu offers four initial subproblems given by four vectors h^1, h^2, h^3, h^4 belonging to the set H. Vectors h^1 and h^2 all have components equal to 1 and -1, respectively. For vector h^3 there is $h_i = (-1)^{i+1}, i = 1, ..., m$, while $h_i = (-1)^i, i = 1, ..., m$ for vector h^4. Geometrically, it means a possibility to start from vertices lying in quite different parts of the polytope X. It can be useful especially for calculating the exact lower bound.

If the initial extremal subproblem has no feasible solution, then the procedure continues by using a systematic procedure to another extremal subproblem by changing exactly one row of the constraints. If all extremal subproblems have no feasible solutions, then the set X of all feasible solutions is empty.

A user's chosen results of outputfiles can be exported into MS Word by means of the option Export.

All the above subroutines were tested in the collection of more than five hundred randomly generated examples. Numerical experience, especially with an implementation of the algorithm LM, could be of some interest. It was confirmed that the algorithm works successfully except for an ILP problem with a nonregular set X. Such a case, however, can be easily recognized during the execution of step 3 of the algorithm. If such a case appears, a warning label is used as cycling may occur. Thus, the algorithm LM terminates after a finite number of steps with a warning of nonregularity, or with satisfying optimality criterion (5), i.e., with finding the exact lower bound in a stable region with the current basis B. There are several possibilities to proceed in a irregular case:

(i) We can try to proceed to another extremal subproblem if condition (5) is not satisfied for more indices.

(ii) We can start the procedure with another initial extremal subproblem.

(iii) We can continue carefully using the algorithm INF while checking the number of iterations to prevent cycling.

To solve irregular problems, procedure (ii) was usually used. There were some problems among the tested collection with the following result: There was an irregularity warning for a certain choice of an initial subproblem, while the algorithm LM terminated succesfully for another choice of the initial subproblem. This has even happened with problems having a relative error $\epsilon < 0.01$.

Regarding the AS algorithm, it can also be used as an examination of some properties of an ILP problem which can be characterized in terms of extremal subproblems, e.g., regularity, feasibility, strong solvability, basic stability and boundedness. A possible application of this type was described briefly in Section 2. The theoretical results mentioned in [6], which are supported by the numerical experience, lead to the conclusion that the algorithms seem to be more useful than other approaches based on the simplex method.

References

1. Beeck, H.: Linear Programming with Inexact Data. Report TUM-ISU-7830, Technical University, Munich (1978).
2. Bunday, B., Garside, G.: Linear Programming in Pascal. E. Arnold (publisher), printed by J. Arrowsmith, Bristol (1987).
3. Jansson, C.: A Self-validating Method for Solving Linear Programming Problems with Interval Input Data. Computing Suppl. 6, 33-46 (1988).
4. Krawczyk, R.: Fehlerabschätzung bei linearer Optimierung. In: Interval Mathematics (K. Nickel, Ed.), Springer-Verlag, Berlin 1975, 215-222.

5. Mráz, F.: On Supremum of the Solution Function in Linear Programs with Interval Coefficients. KAM-Series No. 93-236, Dept. of Applied Math., Charles University, Prague, 1-18 (1993).
6. Mráz, F.: The Algorithm for Solving Interval Linear Programs and Comparison with Similar Approaches. KAM-Series No.93-239, Dept. of Applied Math., Charles University, Prague, 1-11 (1993).
7. Mráz, F.: The Exact Lower Bound of Optimal Values in Interval LP. In: G. Alefeld, A. Frommer and B. Lang (eds.): Scientific Computing and Validated Numerics, Akademie Verlag, Berlin, 214-220 (1996).
8. Oettli, W., Prager, W.: Compatibility of Approximate Solution of Linear Equations with Given Error Bounds for Coefficients and Right-hand Sides. Numerische Mathematik 6, 405-409 (1964).
9. Rohn, J.: Duality in Interval Linear Programming, in: Interval Mathematics (K. Nickel, Ed.), Academic Press, New York (1980).
10. Rohn, J.: Interval Linear Systems. Freiburger Intervall-Berichte 84/7, 33-58 (1984).
11. Rohn, J.: Miscellaneous Results on Linear Interval Systems. Freiburger Intervall-Berichte 85/9, 29-43 (1985).
12. Rohn, J.: NP-Hardness Results for Some Linear and Quadratic Problems. Technical report No. 619, Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague, 1-11 (1995).