In dieser Reihe sind bisher erschienen:

Bericht 93.1 (Januar 1993)
S. M. Rump: Validated Solution of Large Linear Systems

Bericht 93.2 (Januar 1993)
D. Husung: TPX Version 1.1 US Precompiler for Turbo Pascal 5.0

Bericht 93.3 (Juli 1993)
O. Knüppel: BIAS — Basic Interval Arithmetic Subroutines

Bericht 93.4 (Juli 1993)
O. Knüppel: PROFIL — Programmer's Runtime Optimized Fast Interval Library

Bericht 93.5 (November 1993)
O. Knüppel, T. Simenec: PROFIL/BIAS Extensions

Bericht 93.6 (November 1993)
O. Knüppel: A Multiple Precision Arithmetic for PROFIL

Bericht 93.7 (Dezember 1993)
S. M. Rump: Verification Methods for Dense and Sparse Systems of Equations

Bericht 94.1 (Februar 1994)
C. Jansson, O. Knüppel: Numerical Results for a Self-Validating Global Optimization Method

Bericht 94.2 (Juni 1994)
C. Jansson: On Self-Validating Methods for Optimization Problems

Bericht 94.3 (Dezember 1994)
C. Jansson: Some Properties of Linear Interval Systems and Applications

Bericht 95.1 (März 1995)
S. M. Rump: Improved Iteration Schemes for Validation Algorithms for Dense and Sparse Nonlinear Systems

Bericht 95.2 (März 1995)
S. M. Rump: Expansion and Estimation of the Range of Nonlinear Functions

Bericht 95.3 (Mai 1995)
S. M. Rump: Bounds for the Componentwise Distance to the Nearest Singular Matrix

Bericht 95.4 (Oktober 1995)
O. Knüppel: A PROFIL/BIAS Implementation of a Global Minimazation Algorithm

Bericht 95.5 (Oktober 1995)
S. M. Rump: Perron-Frobenius like Theorems for not Sign-Restricted Matrices

Bericht 96.1 (Juli 1996)
S. M. Rump: Almost sharp bounds on the componentwise distance to the nearest singular matrix

Bericht 96.2 (August 1996)
S. M. Rump: Theorems of Perron-Frobenius type for matrices without sign restrictions

Bericht 96.3 (November 1996)
K.-H. Zimmermann: Integral Hecke Modules, Integral Generalized Reed-Muller Codes, and Linear Codes

Bericht 96.4 (Dezember 1996)
C. Jansson, J. Rohn: An Algorithm for Checking Regularity of Interval Matrices

Bericht 97.1 (Februar 1997)
K.-H. Zimmermann, G. Ehlers, W. Brandt, F. A. M. Bouchard, J. Diedrichsen, T. Möller: Das ELLA 2000 Mikrocontroller-Projekt

Bericht 97.2 (Juni 1997)
K.-H. Zimmermann: A Class of Double Coset Codes

Bericht 97.3 (November 1997)
C. Jansson: An NP-hardness result for nonlinear systems

Bericht 98.1 (März 1998)
C. Jansson: Construction of convex lower and concave upper bound functions

Bericht 98.2 (Juli 1998)
G. Ehlers, K.-H. Zimmermann: Einführung in den Hardware-Entwurf mit ELLA

Bericht 98.3 (Oktober 1998)
S.M. Rump: Fast and Parallel Interval Arithmetic

Bericht 98.4 (Oktober 1998)
S.M. Rump: INTLAB – Interval Laboratory

Bericht 99.1 (März 1999)
O. Knüppel: PROFIL/BIAS V 2.0

Bericht 99.2 (März 1999)
J.-P. M. Zemke: b4m – A free interval arithmetic toolbox based on BIAS

Bericht 99.3 (Oktober 1999)
N. Suresh Babu, Karl-Heinz Zimmermann: A Short Introduction to Quaternary Codes

Bericht 99.4 (Oktober 1999)
C. Jansson: Quasiconvex Relaxations Based on Interval Arithmetic

Bericht 00.1 (Oktober 2000)
Karl-Heinz Zimmermann: DNA Algorithms for Binary Linear Codes

Bericht 01.1 (Januar 2001)
Karl-Heinz Zimmermann: The Patch Selection Problem and Discrete Stochastic Dynamic Programming

Bericht 01.2 (März 2001)
Tangha Tina Lai, Karl-Heinz Zimmermann: A Software Platform for the Sticker Model

Bericht 01.3 (Juni 2001)
Prashant Batra: Omitted Values and Real Robust Schur Stability

Bericht 02.1 (Juni 2002)
C. Jansson: Rigorous Lower and Upper Bounds in Linear Programming

Technische Universität Hamburg-Harburg

Berichte des
Forschungsschwerpunktes
Informations- und
Kommunikationstechnik

C. Jansson

Rigorous Lower and Upper Bounds in Linear Programming

Juni 2002

Bericht 02.1

Rigorous Lower and Upper Bounds in Linear Programming

Christian Jansson

June 20, 2002

Inst. of Computer Science III, Technical University Hamburg-Harburg,
Schwarzenbergstraße 95, 21071 Hamburg, Germany. jansson@tu-harburg.de

Keywords: linear programming, interval arithmetic, rigorous error bounds, sensitivity analysis
AMS Subject classification: 90C05, 65G30, 65N15

Abstract

In this note we consider the computation of rigorous lower and upper error bounds for the optimal value of linear programming problems. All input data of the lp-problem may be uncertain and vary between given lower and upper bounds. The error bounds are verified for the family of lp-problems which correspond to these uncertainties. They can be used as well to perform a sensitivity analysis, provided the width of the uncertainties is not too large. Some numerical examples are presented.

1 Introduction

Linear programming problems have been solved with highly large success during the last decades, and many efficient algorithms have been developed. However, on a computer due to rounding errors and/or uncertainties in the input data the computed approximations may be sometimes unsatisfactory. Especially for ill-conditioned problems the results may be wrong.

Ill-conditioning is not a rare phenomenon. In the recent paper of Ordóñez and Freund [16] it is stated that 72% of the lp-instances in the NETLIB Linear Programming Library [12] are ill-conditioned. After applying CPLEX 7.1 presolve (a preprocessing heuristic for linear programming) 19% maintain the property of being ill-conditioned. The NETLIB library contains many industrial problems, and therefore the computation of rigorous error bounds can be valuable in practice.

Beeck [2], Krawzcyk [9], and Rump [19] have developed methods for computing rigorous error bounds for lp-problems where the optimal solution is unique and not degenerate. The common basic idea is to use the simplex method for the computation of an optimal basic index set. Then with interval methods the optimality of this index set is verified a posteriori, and rigorous error bounds for the optimal vertex and the optimal value are calculated. Moreover, these algorithms can be also applied to the calculation of bounds for the family of lp-problems where all real input data vary between given lower and upper bounds. These error bounds require $O(n^3)$ operations, where n denotes the number of variables of the linear programming problem. In Jansson [6] also the degenerate case is considered, by applying the well-known graph search method to an appropriately defined graph of degenerate basic index sets. For each basic index set a linear interval system must be solved, and for all optimal vertices rigorous error bounds are computed. The computational work is $k \cdot O(n^3)$, where k is greater or equal to the number of (degenerate) optimal basic index sets.

The approaches above have in common that for each lp-problem of the family, the existence of optimal solutions is proved. At first, error bounds for the optimal vertices are computed by solving linear interval systems, and then these bounds are used for the computation of error bounds for the optimal value.

The major emphasis in this paper is on the fast computation of rigorous error bounds for the optimal value without using or computing bounds for the optimal vertices. The algorithm presented here avoids to solve linear interval systems, and needs $O(n^2)$ operations for the lower bound, provided that all variables are bounded, and an approximate optimal solution is known. Nothing is assumed about the quality of this approximate solution. The computational work for the upper bound requires $O(mn + p^3)$ operations, where m is the number of inequalities and p is the number of equations of the lp-problem, and beyond it the computational costs for applying the lp-solver to a slightly pertubated problem using the known approximate solution as a starting point.

In the following, we do not consider the standard linear programming problem. The reasons are that transformations to the standard problem may lead to more computational work, and they cause data dependencies in the input data, which may lead to worse error bounds. Therefore, the inequalities are not transformed into equations, and in many cases p is small compared to mn, yielding only $O(mn)$ operations for the upper bound.

It turns out that the lower and the upper bound for the optimal value can also be computed for degenerate and large (sparse) problems, provided the lp-routine in use is suited for these problems. Moreover, the lower bound is computable even for ill-posed problems, whereas the upper bound needs a well-posed property. In return for large values of the interval radius and large simple bounds the methods presented in [2], [9], and [19] compute in many cases better rigorous bounds provided that (i) the lp-problems within the interval input data are well-posed and have unique nondegenerate optimal solutions, and (ii) the larger $n \times n$ interval system coresponding to the optimal basic index set can be solved with narrow bounds.

Rigorous error bounds are not only useful for problems which can be modelled as lp-problems. They can be used for example in global optimization and mixed integer programming, whenever linear relaxations must be solved. Independently and at the same time Neumaier and Shcherbina [15] have developed results which overlap in part with results presented here. However, there are several differences. Their emphasis is more on branch and cut methods for linear mixed integer problems, whereas we discuss problems with uncertain input data, simple bounds which may be infinite, and a different upper bound.

The paper is organized as follows. In Section 2 examples are presented where a commercial lp-solver computes wrong approximations, although the problems are simple. Section 3 contains notation, and in the following section some comments about solving linear interval systems are given. In Section 5 an algorithm for computing a rigorous upper bound of the global minimum value is considered. Then, in Section 6 a rigorous lower bound is presented. After some applications in Section 7, we present numerical results in Section 8. Finally, in Section 9 some conclusions are given.

2 Examples

For many practical lp-problems the input data are integer (and therefore exactly representable on a computer), the problem is well-conditioned, and together with an appropriate pivoting rule (for example Bland's rule) the optimal solution can be computed in a finite number of steps, even in the degenerate case. In addition with the linearity of the problem, it may be argued that these strong assumptions should imply that the computed approximation is satisfactory. But the following example shows that the commercial lp-solver *linprog* in MATLAB [10] fails, even though all above assumptions are satisfied.[1] The problem

$$\begin{array}{rcrcrl} \min -5x_1 & - & 4x_2 & - & 6x_3 & \text{subject to} \\ x_1 & - & 6x_2 & + & x_3 & \leq 20, \\ 3x_1 & + & 2x_2 & + & 4x_3 & \leq 42, \\ 3x_1 & + & 2x_2 & & & \leq 30, \end{array} \tag{1}$$

with $0 \leq x_i \leq \overline{x}_i$ for $i = 1, 2, 3$, is bounded from above by the value 21 for every variable; this can be seen by a short inspection of the second inequality. The optimal solution is $x^* = (0, 15, 3)^T$ with optimal value $f^* = -78$. Hence, the three upper bounds $\overline{x} = (30, 30, 30)^T$, $\overline{x} = (10^{10}, 10^{10}, 10^{10})^T$ or $\overline{x} = (+\infty, +\infty, +\infty)^T$ do not change the set of feasible solutions or the optimal solution. The MATLAB routine *linprog* computes in all three cases the optimal solution and the optimal value with accuracy of about eleven decimal digits. Now, to the above constraints, we add the linear equation

$$x_1 + x_2 + x_3 = 10, \tag{2}$$

and apply *linprog* for this problems with the previous three upper bounds. Notice, that the resulting set of feasible solutions has a nonempty relative interior, and for all three upper bounds the optimal solution is $x^* = (0, 0, 10)$ with optimal value $f^* = -60$.

For the upper bounds $\overline{x} = (30, 30, 30)^T$ and $\overline{x} = (+\infty, +\infty, +\infty)^T$ *linprog* calculates the optimal solution, but surprisingly for the upper bound $\overline{x} = (10^{10}, 10^{10}, 10^{10})$ the message

both the primal and the dual appear to be infeasible

is displayed. The routine *linprog* provides also in the case of a warning the last iteration point before the routine stops. For this example, this point is equal to $x = (0.0045, 5.6042, 10.4218)$, which does not satisfy equation (2).

In the second example, we tried to solve with *linprog* the problem

$$\begin{array}{rccl} \min & \sum_{i=1}^{n} & s_i & \text{subject to} \\ Ax & + & s & = e_1, \\ 0 & \leq x_i, & s_i & \leq 1000 \text{ for } i = 1 \cdots n, \end{array} \tag{3}$$

where $x, s \in \mathbf{R}^n$, e_1 is the first unit vector, and A is the inverse of the $n \times n$ Pascal matrix. Notice that the Pascal matrix and its inverse have integer entries. A short calculation shows that the optimal solution is $x = A^{-1}e_1$ (which is 1 in every component of x), and $s = 0$. The basis matrix corresponding to the optimal solution is A. For $n = 9$ the 2-norm condition number of A is 2.9×10^8. Because MATLAB uses double precision, one would expect that about 8 decimal places of the optimal solution are correct. In fact, this is true if the solution of the linear system $Ax = e_1$ is computed by the linear system solver in MATLAB. However, the MATLAB-routine *linprog* displays the message

the primal appears to be infeasible (and the dual unbounded).

We mention that for our first example the results of MATLAB Version 5.3 and Version 6.0 are identical. For the second example they differ slightly. The message above comes from Version 5.3. Version 6.0 displays after 86 iterations the message

Maximum number of iterations exceeded,

[1] By accident, this example was discussed during a course of lectures about optimization for students studying electrical engineering. During this course, branch and bound strategies for linear mixed integer problems and the influence of the simple bounds were investigated. We used the MATLAB routine *linprog*, and surprisingly by increasing the upper bounds the set of feasible solutions became empty. Unfortunately, I cannot find the reference for the data of this lp-problem. It could be that it was originally presented in a book borrowed per interlibrary loan.

and the component $x(9)$ of x in the last iteration is 4.858×10^{-15}, even though the optimal component is one. Increasing the maximum number of iterations to 1000 does not change anything. For $n = 10$ the 2-norm condition number of A is 4.2×10^9, and also in Version 6.0 the message

the primal appears to be infeasible (and the dual unbounded),

is displayed.

A third example can be found in Neumaier and Shcherbina [15]. They give an innocent-looking mixed linear integer problem where the commercial, high quality solver CPLEX [5] and several others failed. The reason is that the linear programming relaxations are not solved correctly by the lp-algorithm.

In backward error analysis it is described that many algorithms are backward stable; that is, the computed solution is the exact result of a problem with slightly pertubated input data. A main assumption in most proofs for backward stability of algorithms is that the pertubations of the input data vary componentwise independently. But this assumption is not fulfilled for many real life problems.

A short inspection of our examples shows that *linprog* is not backward stable. But even backward stable results may be unsatisfactory in applications. This is pointed out by Neumaier and Shcherbina [15] who writes:

> *However, backward error analysis has no relevance for integer linear programs with integer coefficients, since slightly pertubated coefficients no longer produce problems of the same class.*

Algorithms for solving optimization problems search for an optimal solution. It is not surprising that sometimes the optimum cannot be found, although a convergence theory is well-elaborated and a well-known termination criteria is implemented. However, a linkage of search algorithms with rigorous error bounds makes it possible to obtain mathematically correct results for the optimal value of linear programming problems with limited computational work. It turns out that the computation of error bounds can be added a-posteriori in form of a subroutine at the end of each linear programming code. Even in some cases (see the following examples) where unsatisfactory approximations are computed, useful bounds can be obtained.

3 Notation

Throughout this paper we use the following notation. $\mathbf{R}$, $\mathbf{R}^n$ and $\mathbf{R}^{m\times n}$ denote the sets of real numbers, real vectors, and real $m \times n$ matrices, respectively. Comparisons $\leq$ and absolute value $|.|$ are used entrywise. The coefficients of a real $m \times n$ matrix A are denoted by A_{ij}, its columns by $A_{:j}$, its rows by $A_{i:}$, and its transpose by A^T.

We require only some basic definitions about interval arithmetic which are described in this note. There are a number of textbooks on interval arithmetic and self-validating methods which can be highly recommended to readers. These include Alefeld and Herzberger [1], Kearfott [7], Moore [11], and Neumaier [13], [14].

If $\mathbf{V}$ is one of the spaces $\mathbf{R}$, $\mathbf{R}^n$, $\mathbf{R}^{m\times n}$, and $\underline{v}, \overline{v} \in \mathbf{V}$, then the box

$$\mathbf{v} := [\underline{v}, \overline{v}] := \{v \in \mathbf{V} : \ \underline{v} \leq v \leq \overline{v}\} \tag{4}$$

is called an *interval quantity* in $\mathbf{V}$ with *lower bound* $\underline{v}$ and *upper bound* $\overline{v}$. In particular, $\mathbf{IR}$, $\mathbf{IR}^n$, and $\mathbf{IR}^{m\times n}$ denote the set of real intervals $\mathbf{a} = [\underline{a}, \overline{a}]$, the set of real interval vectors $\mathbf{x} = [\underline{x}, \overline{x}]$, and the set of real interval matrices $\mathbf{A} = [\underline{A}, \overline{A}]$, respectively. The real operations $A \circ B$ with $\circ \in \{+, -, \cdot, /\}$ between real numbers, real vectors and real matrices can be generalized to interval operations. The result $\mathbf{A} \circ \mathbf{B}$ is defined as the interval hull of all possible real results, that is

$$\mathbf{A} \circ \mathbf{B} := \bigcap \{\mathbf{C} \in \mathbf{IV} : \ A \circ B \in \mathbf{C} \quad \text{for all} \quad A \in \mathbf{A}, B \in \mathbf{B}\}. \tag{5}$$

All interval operations can be easily executed by working appropriately with the lower and upper bounds of the interval quantities. For example, in the simple case of addition, we obtain

$$\mathbf{A} + \mathbf{B} = [\underline{A} + \underline{B}, \overline{A} + \overline{B}]. \tag{6}$$

For interval quantities $\mathbf{A}, \mathbf{B} \in \mathbf{IV}$ we define

$$\begin{aligned}
\check{\mathbf{A}} &:= (\underline{A} + \overline{A})/2 \quad \text{as the } \textit{midpoint}, && (7)\\
\mathrm{rad}(\mathbf{A}) &:= (\overline{A} - \underline{A})/2 \quad \text{as the } \textit{radius}, && (8)\\
|\mathbf{A}| &:= \sup\{|A| : A \in \mathbf{A}\} \quad \text{as the } \textit{absolute value}, && (9)\\
\mathbf{A}^+ &:= \{A \in V : A \in \mathbf{A},\, A \geq 0\}, && (10)\\
\mathbf{A}^- &:= \{A \in V : A \in \mathbf{A},\, A \leq 0\}. && (11)
\end{aligned}$$

Moreover, the comparison in $\mathbf{IV}$ is defined by

$$\mathbf{A} \leq \mathbf{B} \quad \text{iff} \quad \overline{A} \leq \underline{B},$$

and other relations are defined analogously. Real quantities v are embedded in the interval quantities by identifying $v = \mathbf{v} = [v, v]$.

4 Linear Interval Systems

Linear systems of equations with inexact input data are treated in interval arithmetic by working with an interval matrix $\mathbf{A} \in \mathbf{IR}^{n\times n}$ and a right hand side $\mathbf{b} \in \mathbf{IR}^n$. The aim is to compute an interval vector $\mathbf{x} \in \mathbf{IR}^n$ containing the *solution set*

$$\Sigma(\mathbf{A}, \mathbf{b}) := \{x \in \mathbf{R}^n : Ax = b \text{ for some } A \in \mathbf{A},\, b \in \mathbf{b}\} \tag{12}$$

If all $A \in \mathbf{A}$ are nonsingular, then finding an enclosure $\mathbf{x}$ of the solution set is equivalent to the property

$$\forall\, A \in \mathbf{A},\ \forall\, b \in \mathbf{b}\ \exists\, x \in \mathbf{x} :\ Ax = b. \tag{13}$$

This is an NP-hard problem, but there are several methods that compute enclosures $\mathbf{x}$ with $O(n^3)$ operations for certain types of interval matrices. A precise description of such methods, required assumptions, and approximation properties can be found for example in Neumaier [13]. Roughly speaking, it turns out that for interval matrices with $\|I - R\mathbf{A}\| < 1$ (R is an approximate inverse of the midpoint $\check{\mathbf{A}}$) there are several methods which compute an enclosure $\mathbf{x}$, and the radius $\mathrm{rad}(\mathbf{x})$ decreases linearily with decreasing radii $\mathrm{rad}(\mathbf{A})$ and $\mathrm{rad}(\mathbf{b})$. For the computation of enclosures in the case of large linear systems the reader is referred to Rump [20].

The computation of rigorous lower and upper bounds for the optimal value requires to consider a modified problem. There, a nonsquare interval matrix $\mathbf{A} \in \mathbf{IR}^{m\times n}$ with $m < n$ and a right hand side $\mathbf{b} \in \mathbf{IR}^m$ are given, and the aim is to compute an interval vector $\mathbf{x} \in \mathbf{IR}^n$ such that property (13) is fulfilled.

Since $m < n$ (in most cases m is much smaller than n), the solution set $\Sigma(\mathbf{A}, \mathbf{b})$ is in almost all cases unbounded, whereas property (13) requires to find only an enclosure of a part of the solution set.

Obviously, there are many possibilities for computing such a part of the solution set. We need to compute such an enclosure $\mathbf{x}$ with respect to a given vector $\tilde{x}$ and an index set $I := (\beta_1, \ldots, \beta_m) \subseteq \{1, ..., n\}$, and proceed as follows:

1. Set $\mathbf{A}_{:I} := (\mathbf{A}_{:\beta_1}, \ldots, \mathbf{A}_{:\beta_m})$, and denote the remaining matrix of $\mathbf{A}$ by $\mathbf{A}_{:N} := (\mathbf{A}_{:\gamma_1} \ldots, \mathbf{A}_{:\gamma_{n-m}})$, where $N := \{\gamma_1, \ldots, \gamma_{n-m}\} = \{1, \ldots, n\}\backslash I$.

2. Set $\mathbf{x}_{\gamma_1} := \widetilde{x}_{\gamma_1}, \ldots, \mathbf{x}_{\gamma_{n-m}} := \widetilde{x}_{\gamma_{n-m}}$.
3. Compute an enclosure $(\mathbf{x}_{\beta_1}, \ldots, \mathbf{x}_{\beta_m})^T \in \mathbf{IR}^m$ of the solution set with square interval matrix $\mathbf{A}_{:I}$ and right hand side

$$\mathbf{b} - \sum_{j=1}^{n-m} \mathbf{A}_{:\gamma_j} \widetilde{x}_{\gamma_j} \tag{14}$$

by using an algorithm for square linear interval systems.

This algorithm yields for every $A \in \mathbf{A}$ and $b \in \mathbf{b}$ a real vector $x \in \mathbf{x}$ with

$$x_i \in \mathbf{x}_{\beta_j} \ \text{ if } \ i = \beta_j, \text{ and } x_i = \widetilde{x}_{\gamma_j} \text{ if } i = \gamma_j \tag{15}$$

such that $Ax = b$, i.e. property (13) is satisfied, and the part of the solution set matching with $\widetilde{x}_{\gamma_j}$ for all $\gamma_j \in N$ is filtered out. Because the components $\widetilde{x}_{\gamma_j}$ are real, the right hand side (14) can be computed without overestimation.

5 A Rigorous Upper Bound for the Optimal Value

In this section we investigate the computation of a rigorous upper bound for the optimal value and the certificate of existence of optimal solutions for linear programming problems. Many economic or technical variables are bounded from below by zero, and in most cases reasonable upper bounds can be given. Therefore, our major focus is on bounded lp-problems. However, we allow also unbounded variables.

Frequently, the transformations to the standard lp-problem lead to data dependencies of the interval input data. For example, describing an unsigned variable as the difference of two nonnegative variables introduces two dependent columns into the system matrix. For interval input data such dependencies lead to overestimations.

Therefore, we consider the nonstandard problem

$$\min_{x \in F} c^T x, \quad F := \{x \in \mathbf{R}^n : \ Ax \le a, \ Bx = b, \ \underline{x} \le x \le \overline{x}\}, \tag{16}$$

where $A \in \mathbf{R}^{m \times n}$, $B \in \mathbf{R}^{p \times n}$, $c, x \in \mathbf{R}^n$, $a \in \mathbf{R}^m$, and $b \in \mathbf{R}^p$.

F is called the set of *(primal) feasible solutions*, F^* denotes the set of *optimal solutions*, and f^* is the *optimal value*. If F is empty, the $f^* := +\infty$. This problem is described by the input data

$$P = (A, B, a, b, c) \in \mathbf{R}^{(m+p+1)n+m+p}, \tag{17}$$

and the simple bounds $\underline{x} < \overline{x}$, which may be infinite; that is $\underline{x}_j := -\infty$ or $\overline{x}_j := +\infty$ for some $j \in \{1, \ldots, n\}$, and following we use the arrangement $0 \cdot (+\infty) = 0 \cdot (-\infty) = 0$. The set of indices where the simple bounds are both infinite is denoted by

$$J^\infty := \{j \in \{1, \ldots, n\} : \ \underline{x}_j = -\infty, \text{and } \overline{x}_j = +\infty\}, \tag{18}$$

and its complement is

$$J^r := \{1, \ldots, n\} \setminus J^\infty. \tag{19}$$

In many applications J^∞ is empty or contains only few indices. A feasible point $x \in F$ is called *degenerate*, if more than n constraints in F are active, that is at least $n + 1$ constraints hold as equations in x.

Some or all input data of a linear programming problem may be uncertain. We describe these uncertainties by considering a family of lp-problems P, where $P \in \mathbf{P}$ and

$$\mathbf{P} := (\mathbf{A}, \mathbf{B}, \mathbf{a}, \mathbf{b}, \mathbf{c}) \in \mathbf{IR}^{(m+p+1)n+m+p}. \tag{20}$$

To indicate the dependency of the notation above from $P \in \mathbf{P}$, we write sometimes $F(P), F^*(P), f^*(P), x(P)$, etc.

One possibility to obtain a rigorous error bound, is to use a similar approach as described in Beeck [2], Krawzcyk [9], and Rump [19]. First, we compute an enclosure $\mathbf{x}$ of the solution set for the $n \times n$ linear interval system corresponding to the optimal basic index set which is computed by the lp-solver. Secondly, we check whether all remaining inequalities and simple bound conditions hold true for $\mathbf{x}$. If this is the case, then each problem $P \in \mathbf{P}$ has a primal feasible solution in $\mathbf{x}$, and and we obtain the upper bound $\overline{f}^* := \sup(\mathbf{c}^T\mathbf{x})$. For degenerate or nearly degenerate problems this approach fails, because $\mathbf{x}$ does not satisfy the remaining inequalities.

In the following, we want to calculate upper bounds also for degenerate well-posed problems. It turns out that a slightly perturbated lp-problem and a $p \times p$ linear interval system must be solved. For linear programming problems which are defined only by inequalities $p = 0$, and no interval system needs to be considered. The basic idea is the determination of an interval vector $\mathbf{x}$ which contains for every lp-problem of the family a relative interior primal feasible solution. This solution must be close to an optimal solution, but sufficiently far away from degeneracy and infeasibility. The next theorem gives favourable characteristics of $\mathbf{x}$.

Theorem 5.1 *Let $\mathbf{P} := (\mathbf{A}, \mathbf{B}, \mathbf{a}, \mathbf{b}, \mathbf{c})$ be a family of lp-problems with input data $P \in \mathbf{P}$ and simple bounds $\underline{x} < \overline{x}$. Suppose that there exists an interval vector $\mathbf{x} \in I\!\mathbf{R}^n$ such that*

$$\mathbf{A}\mathbf{x} \le \mathbf{a}, \ \underline{x} \le \mathbf{x} \le \overline{x}, \tag{21}$$

and

$$\forall\, B \in \mathbf{B},\ \forall\, b \in \mathbf{b}\ \exists\, x \in \mathbf{x} :\ Bx = b. \tag{22}$$

Then for every $P \in \mathbf{P}$ there exists a primal feasible solution $x(P) \in \mathbf{x}$, and the inequality

$$\sup_{P \in \mathbf{P}} f^*(P) \le \overline{f}^* := \sup(\mathbf{c}^T\mathbf{x}) \tag{23}$$

is satisfied. Moreover, if the objective function is bounded from below for every lp-problem with input data $P \in \mathbf{P}$, then each problem has an optimal solution.

Proof. Let $P = (A, B, a, b, c) \in \mathbf{P}$ be a fixed chosen problem. The condition (22) implies that there exists a $x(P) \in \mathbf{x}$ with $B \cdot x(P) = b$, and from condition (21) it follows that $x(P)$ is a primal feasible solution. Hence, $c(P)^T x(P) \ge f^*(P)$, and (23) follows. If the objective function $c(P)^T x$ is bounded from below by a finite number $\underline{f}^*$, then the theory of linear programming proves the existence of optimal solutions for P. ∎

In the next section we show how a rigorous lower bound for the optimal value $\underline{f}^*$ can be calculated. Now, it remains to describe the algorithm for computing a rigorous upper bound $\overline{f}^*$ and an appropriate interval vector $\mathbf{x}$ satisfying the conditions (21) and (22). We assume that an approximate solution $\widetilde{x}$ of the midpoint problem $\check{\mathbf{P}}$ is known. Nothing is assumed about the quality of $\widetilde{x}$. The algorithm consists of the following seven steps:

(1) Let ε be the adjusted accuracy of the lp-solver, let η be a number greater than the smallest positive floating point number, let $1.5 \le \alpha \le 5$, and let e be the vector which is equal 1 in every component. Compute the *deflation parameters*

$$\varepsilon_a := \alpha(\mathrm{rad}(\mathbf{a}) + \mathrm{rad}(\mathbf{A})|\widetilde{x}| + \varepsilon(|a| + |A||\widetilde{x}|)) + \eta \cdot e \tag{24}$$

and $\underline{\varepsilon}_s, \overline{\varepsilon}_s$ which are defined componentwise by

$$(\underline{\varepsilon}_s)_j := \begin{cases} 0 & \text{if } \underline{x}_j = -\infty \\ \varepsilon|\underline{x}_j| + \eta & \text{otherwise,} \end{cases} \tag{25}$$

$$(\overline{\varepsilon}_s)_j := \begin{cases} 0 & \text{if } \overline{x}_j = +\infty \\ \varepsilon|\overline{x}_j| + \eta & \text{otherwise.} \end{cases} \tag{26}$$

(2) Define the pertubated problem $P(\varepsilon) := (\check{\mathbf{A}}, \check{\mathbf{B}}, \check{\mathbf{a}} - \varepsilon_a, \check{\mathbf{b}}, \check{\mathbf{c}})$ with simple bounds $\underline{x}(\varepsilon) := \underline{x} + \underline{\varepsilon}_s$ and $\overline{x}(\varepsilon) := \overline{x} - \overline{\varepsilon}_s$.

(3) Compute an approximate optimal solution $\tilde{\tilde{x}}$ of the pertubated problem $P(\varepsilon)$ using the starting point $\tilde{x}$.

If the lp-solver cannot find an approximate feasible solution, then STOP: *No upper bound for the optimal value.*

(4) Redefine $\tilde{\tilde{x}}$; that is, every component of $\tilde{\tilde{x}}$ being smaller than the corresponding component of $\underline{x}(\varepsilon)$ is set equal to this lower bound, and every component of $\tilde{\tilde{x}}$ being larger than the corresponding component of the simple bound $\overline{x}(\varepsilon)$ is set equal to this upper bound. Then $\underline{x}(\varepsilon) \le \tilde{\tilde{x}} \le \overline{x}(\varepsilon)$.

Set $\tilde{x} := \tilde{\tilde{x}}$.

(5) For $p = 0$ set $\mathbf{x} := \tilde{x}$.

Otherwise, compute $r := |\tilde{x} - (\underline{x}(\varepsilon) + \overline{x}(\varepsilon))/2|$, and let the index set $I = \{\beta_1, \ldots, \beta_p\}$ consist of the p indices with smallest values r_{β_i}. For the nonsquare linear interval system with input data $\mathbf{B}, \mathbf{b}$ compute with the algorithm of Section 4 with respect to $\tilde{x}$ and I the interval vector $\mathbf{x}$ such that condition (22) is fulfilled.

If this algorithm does not compute an enclosure $\mathbf{x}$, then STOP: *No upper bound for the optimal value.*

(6) If the conditions $\mathbf{Ax} \le \mathbf{a}$, $\underline{x} \le \mathbf{x} \le \overline{x}$ are satisfied, then STOP: *Upper bound* $\overline{f}^* = \sup(\mathbf{c}^T\mathbf{x})$.

(7) Increase the deflation parameters by setting $\varepsilon_a := \alpha\varepsilon_a, \underline{\varepsilon}_s := \alpha(\underline{\varepsilon}_s + \mathrm{rad}(\mathbf{x})), \overline{\varepsilon}_s := \alpha(\overline{\varepsilon}_s + \mathrm{rad}(\mathbf{x}))$.

(8) If the lp-solver gives no warning then goto step (2).

It follows that in each iteration (i.e. one execution of steps (2) to (8)) the deflation parameters $\varepsilon_a, \underline{\varepsilon}_s, \overline{\varepsilon}_s$ are increased relatively and absolutely by adding positive values $\mathrm{rad}(\mathbf{x})$ and η. This implies that in each iteration the set of primal feasible solutions $F(P(\varepsilon))$ is contained in the relative interior of the set of feasible solutions of the previous iteration. Hence, these sets are shrinking in each iteration at most until they become empty. In this case the algorithm terminates in step(3), because the lp-solver cannot find an approximate optimal solution. Therefore, the algorithm terminates either in step(6) with calculating an appropriate $\mathbf{x}$, or the algorithm terminates in one of the steps(3) or (5). If the algorithm terminates in step(5), then one may choose another method for solving linear interval systems, or the radius $\mathrm{rad}(\mathbf{P})$ must be decreased. In our present implementation of the algorithm we stop after at most 10 iterations.

In the case of sufficiently small $\mathrm{rad}(\mathbf{P}), \varepsilon, \eta$, and under the assumption that the lp-solver and the solver for linear interval systems work appropriately , the algorithm computes an upper bound $\overline{f}^*$ if $F(\check{\mathbf{P}})$ has a nonempty relative interior, i.e. the real linear system

$$\check{\mathbf{A}}x < \check{\mathbf{a}}, \check{\mathbf{B}}x = \check{\mathbf{b}},\ \underline{x} < x < \overline{x} \tag{27}$$

has a solution x. The property that the relative interior of system (27) is nonempty remains true for small data pertubations, that is the set of feasible solutions must be well-posed. On the other hand, degeneracy does not affect the practicability of this algorithm, because we search for an interval vector contained in the relative interior of F. However, the number of iterations may increase for degenerate problems.

It is not required that the pertubated problem $P(\varepsilon)$ is contained in $\mathbf{P}$, but the crucial property is that the conditions for $\mathbf{x}$ in step(6) hold true. This latter property implicitely demands that the deflation parameters are chosen such that the set of feasible solutions $F(P(\varepsilon))$ is contained in the relative interior of $F(P)$ for every $P \in \mathbf{P}$.

The deflation parameters affect the computational work and the error bound. For example, large values of α decrease the number of iterations, but the quality of the bound becomes worse because $\mathbf{x}$ will be removed towards the relative interior of $F(\check{\mathbf{P}})$ and away from optimality. In our present experience the value $\alpha = 2.5$ is used. The other parameters $\varepsilon_a, \underline{\varepsilon}_s$ and $\overline{\varepsilon}_s$ are chosen in a heuristic manner in order to comprise the interval input data and the rounding errors.

The index set $I = \{\beta_1, \ldots, \beta_p\}$ in step (5) is chosen such that the components of $\mathbf{x}$ with diameter greater zero are far away from the simple bounds. Then the check $\underline{x} \leq \mathbf{x} \leq \overline{x}$ is easier to satisfy.

Example: For the purpose of illustration we consider a degenerate lp-problem, where the constraints are taken from Vanderbei [22], page 36:

$$\begin{array}{lcccll} \min & -x & -x_2 & -4x_3 & \text{subject to} & \\ & x_1 & & +2x_3 & \leq & 2 \\ & & x_2 & +2x_3 & \leq & 2 \\ 0 \leq & x_1 & , x_2 & , x_3 & \leq & 2 \end{array} \tag{28}$$

which is illustrated in Figure 5.1

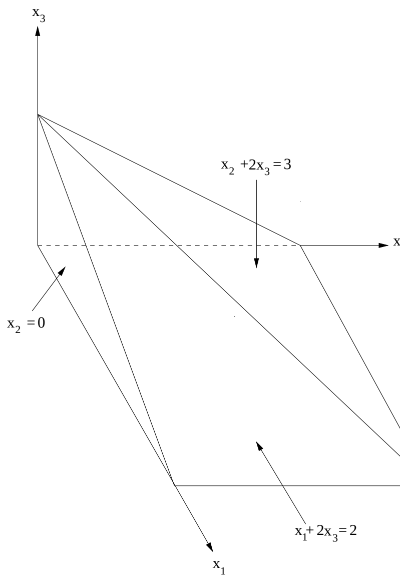

Figure 5.1 *The set of feasible solutions for the degenerate problem.*

Obviously, $x^* = (0,0,1)$ is optimal with value $f^* = -4$. The vector x^* is the intersection of four of the facets, and not of three facets as one would normally expect, i.e. x^* is degenerate. Moreover, it is not a simple degeneracy caused by redundant constraints, since deleting only one of the above constraints changes the set of feasible solutions.

Now, we replace each nonzero coefficient c of the objective function and the constraints by an interval $[c-r, c+r]$ with $0 \le r \le 1/2$, yielding an interval problem $\mathbf{P}$. Then (28) is the midpoint problem of $\mathbf{P}$.

To simplify matters for the following illustration of the previous algorithm, we assume that the lp-solver calculates the exact optimal solution, and we set $\epsilon := 0$, $\eta := 0$, $\alpha = 3/2$ and $\tilde{x} := x^*$. Then in step (1) we obtain

$$\epsilon_a = \frac{3}{2}\left(\begin{pmatrix} r \\ r \end{pmatrix} + \begin{pmatrix} r & 0 & r \\ 0 & r & r \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}\right) = \begin{pmatrix} 3r \\ 3r \end{pmatrix}, \; \underline{\varepsilon}_s = \overline{\varepsilon}_s = 0$$

yielding in step (2) the constraints of the pertubated problem

$$x_1 + 2x_3 \le 2 - 3r, \quad x_2 + 2x_3 \le 2 - 3r.$$

The optimal solution of this perturbated problem is $\tilde{\tilde{x}} = (0, 0, 1 - \frac{3}{2}r)$. In step(5) $p = 0$ and $\mathbf{x} = (0, 0, 1 - \frac{3}{2}r)$. Since $0 \le \mathbf{x} \le 2$, and furthermore the inequality

$$[1-r, 1+r] \cdot 0 + [2-r, 2+r](1 - \frac{3}{2}r) = [2 - 4r + \frac{3}{2}r^2, 2 - 2r - \frac{3}{2}r^2] \le [2-r, 2+r]$$

holds valid for $r \le \frac{1}{2}$, all conditions in step (6) are fulfilled. Hence,

$$\overline{f^*} = \sup([-4-r, -4+r] \cdot (1 - \frac{3}{2}r)) = -4 + 7r - \frac{3}{2}r^2. \tag{29}$$

6 A Rigorous Lower Bound for the Optimal Value

Closely related to problem (16) is the *dual problem*

$$\begin{array}{ll} \max & a^T y + b^T z + \underline{x}^T u + \overline{x}^T v \quad \text{subject to} \\ & A^T y + B^T z + u + v = c \\ & y \le 0, \; u \ge 0, \; v \le 0. \end{array} \tag{30}$$

A vector $(y, z, u, v)^T \in \mathbf{R}^{2n+m+p}$ satisfying the constraints in (30) is called a *(dual) feasible point* , G denotes the set of dual feasible points in (30), G^* is the set of optimal points, and g^* denotes the optimal value. Observe that with our arrangement the objective value of a dual feasible point $(y, z, u, v)^T$ is finite, iff $u_j = 0$ for $\underline{x}_j = -\infty$ and $v_j = 0$ for $\overline{x}_j = +\infty$.

The dual problem is described by the same input data as problem (16), and the theory of linear programming shows that both optimal values f^* and g^* are finite and equal, provided that optimal solutions exist. A feasible solution $(y, z, u, v)^T \in G$ is called *(dual) degenerate*, if less than n of its components are nonzero.

The following theorem provides a rigorous lower bound of $f^*(P)$ for all $P \in \mathbf{P}$ by using the dual problem.

Theorem 6.1 *Let* $\mathbf{P} := (\mathbf{A}, \mathbf{B}, \mathbf{a}, \mathbf{b}, \mathbf{c})$ *be a family of lp-problems with input data* $P \in \mathbf{P}$ *and simple bounds* $\underline{x} < \overline{x}$. *Suppose that there exist interval vectors* $\mathbf{y} \in \mathbf{IR}^m$ *and* $\mathbf{z} \in \mathbf{IR}^p$ *such that*

(i) the sign condition

$$\mathbf{y} \le 0 \tag{31}$$

holds true,

(ii) the equations

$$\forall A \in \mathbf{A}, B \in \mathbf{B}, c \in \mathbf{c}\ \exists y \in \mathbf{y}, z \in \mathbf{z}:$$
$$(A_{:j})^T y + (B_{:j})^T z = c_j \text{ for } j \in J^\infty \tag{32}$$

are fulfilled,

(iii) and further for the intervals

$$\mathbf{d}_j := \mathbf{c}_j - (\mathbf{A}_{:j})^T \mathbf{y} - (\mathbf{B}_{:j})^T \mathbf{z} \text{ for } j \in J^r \tag{33}$$

the inequalities

$$\mathbf{d}_j \leq 0, \text{ if } \underline{x}_j = -\infty \text{ and } j \in J^r, \tag{34}$$
$$\mathbf{d}_j \geq 0, \text{ if } \overline{x}_j = +\infty \text{ and } j \in J^r \tag{35}$$

are satisfied.

Then the inequality

$$\inf_{P \in \mathbf{P}} f^*(P) \geq \underline{f}^* := \min(\mathbf{a}^T \mathbf{y} + \mathbf{b}^T \mathbf{z} + \sum_{\substack{j \in J^r \\ \underline{\mathbf{d}}_j > 0}} \underline{x}_j \mathbf{d}_j^+ + \sum_{\substack{j \in J^r \\ \underline{\mathbf{d}}_j < 0}} \overline{x}_j \mathbf{d}_j^-) \tag{36}$$

is fulfilled, and $\underline{f}^*$ is a finite lower bound of the global minimum value. Moreover, if

(a) all input data are real (i.e. $P = \mathbf{P}$),

(b) P has an optimal solution (y^*, z^*, u^*, v^*),

(c) $\mathbf{y} := y^*$, $\mathbf{z} := z^*$,

(d) the quantities in (33) and (36) are calculated exactly,

then the conditions (i),(ii) and (iii) are satisfied, and the optimal value $f^*(P) = \underline{f}^*$; that is, this lower error bound is sharp for real input data and exact computations.

Proof. Let $\mathbf{y}, \mathbf{z}$ be interval vectors satisfying conditions (i), (ii), and (iii), and let $y \in \mathbf{y}, z \in \mathbf{z}$ be vectors which satisfy equation (32) for a fixed chosen problem $P = (A, B, a, b, c) \in \mathbf{P}$. By definition (33)

$$d_j := c_j - (A_{:j})^T y - (B_{:j})^T z \in \mathbf{d}_j \tag{37}$$

for every $j \in J^r$. We define componentwise for $j = 1, \cdots, n$ two vectors $u, v \in \mathbf{R}^n$ by

$$u_j := \begin{cases} d_j & \text{for } j \in J^r \text{with } d_j > 0 \\ 0 & \text{otherwise,} \end{cases} \tag{38}$$

$$v_j := \begin{cases} d_j & \text{for } j \in J^r \text{with } d_j < 0 \\ 0 & \text{otherwise.} \end{cases} \tag{39}$$

Using (32), (37), (38), and (39), it follows immediately that the vector (y, z, u, v) satisfies

$$(A_{:j})^T y + (B_{:j})^T z + u_j + v_j = c_j \quad \text{for} \quad j = 1, \dots, n.$$

The sign condition (31), and the definitions above yield $y \le 0$, $u \ge 0$, and $v \le 0$, implying that this vector is dual feasible. Hence, using the inequalities and equations (16), we obtain for every primal feasible $x \in F$

$$\begin{aligned}&a^T y + b^T z + \underline{x}^T u + \overline{x}^T v \le \\ &x^T A^T y + x^T B^T z + x^T u + x^T v = \\ &x^T c.\end{aligned}$$

Noting that $u_j = v_j = 0$ for $j \in J^\infty$, it follows that the corresponding terms $\underline{x}_j u_j$ and $\overline{x}_j v_j$ vanish in the dual objective function, and

$$f^*(P) = \min_{x \in F} c^T x, \ge a^T y + b^T z + \sum_{j \in J^r} \underline{x}_j u_j + \sum_{j \in J^r} \overline{x}_j v_j. \tag{40}$$

The bounds (33) yield $d_j \in \mathbf{d}_j$ for every $j \in J^r$, and definitions (38) and (39) imply

$$\begin{aligned}\underline{x}_j u_j &= 0 \quad \text{or} \quad \underline{x}_j u_j \in \underline{x}_j \mathbf{d}_j^+ \quad \text{for} \quad j \in J^r, & (41)\\ \overline{x}_j v_j &= 0 \quad \text{or} \quad \overline{x}_j v_j \in \overline{x}_j \mathbf{d}_j^- \quad \text{for} \quad j \in J^r \quad . & (42)\end{aligned}$$

If we put (41) and (42) into (40), and if we consider all $P \in \mathbf{P}$, we obtain the rigorous lower bound (36). The inequalities (34) and (35) implicate that no infinite terms appear in (36). Hence, the lower bound is finite.

Let the input data be real, and let (y^*, z^*, u^*, v^*) be a dual optimal solution. Then define $\mathbf{y} := y^*$, $\mathbf{z} := z^*$. If the d_j are calculated exactly, then (38), (39) yield $u = u^*$, $v = v^*$ and the constructed vector $(\mathbf{y}, \mathbf{z}, u, v)$ is the optimal vector of the dual problem. A short inspection shows the validity of (i),(ii) and (iii), and the duality theorem of linear programming proves the last assertion. ∎

A consequence is the following error bound for standard linear programming problems with real input data:

Corollary 6.1 *Let $B \in \mathbf{R}^{p \times n}$, $c \in \mathbf{R}^n$, $b, z \in \mathbf{R}^p$, and $d := c - B^T z$. Then the optimal value f^* of the standard lp-problem*

$$\min_{x \in F} c^T x, \quad F := \{x \in \mathbf{R}^n : \ Bx = b, \ 0 \le x \le \overline{x}\} \tag{43}$$

with simple upper bound $\overline{x} \in \mathbf{R}^n$, $0 < \overline{x}$ satisfies the inequality

$$f^* \ge b^T z + \sum_{\substack{j=1 \\ d_j < 0}}^{n} \overline{x}_j \cdot d_j. \tag{44}$$

Proof. This follows from Theorem 6.1 by noticing that A is empty, all simple bounds are finite, and $\underline{x} = 0$. Definition (33) yields the vector d for given z. ∎

Obviously, the bound (44) depends only on the quality of z. If z is close to a dual optimal solution of (43) then f^* is close to the right hand side of (44). This follows immediatedly from the dual problem of (43), which is

$$\max b^T z + \overline{x}^T v \quad \text{subject to} \quad c - B^T z = v, v \le 0. \tag{45}$$

More general, the lower bound $\underline{f}^*$ in Theorem 6.1 depends mainly on the radius r of the interval input data, and the chosen interval vectors $\mathbf{y}$ and $\mathbf{z}$. From the duality theory it follows that the quality of this bound is improved, if $\mathbf{y}$ and $\mathbf{z}$ are close to or contain the corresponding parts y^*, z^* of the dual optimal solution.

At first we discuss the algorithm for computing $\underline{f}^*$ in the case where all simple bounds are finite. Later the general case is treated.

We assume that approximations $\tilde{y}, \tilde{z}$ of a dual optimal solution (y^*, z^*, u^*, v^*) of the midpoint problem of $\mathbf{P}$ are already computed.

(1) Redefine $\widetilde{y}$ by setting all positive components equal to zero.

(2) Set $\mathbf{y} := \widetilde{y}$, and $\mathbf{z} := \widetilde{z}$.

(3) Compute $\mathbf{d} := \mathbf{c} - \mathbf{A}^T\mathbf{y} - \mathbf{B}^T\mathbf{z}$.

(4) Compute $\underline{f}^*$ with formula (36).

Since all simple bounds are finite, the conditions (ii) and (iii) are fulfilled by definition, and because $\mathbf{y} \leq 0$ (see steps(1) and (2)) also condition (i) is satisfied, yielding the rigorous bound $\underline{f}^*$. This bound requires only linear operations between interval quantities. Since $\text{rad}(\mathbf{y}) = \text{rad}(\widetilde{y}) = 0$ and $\text{rad}(\mathbf{z}) = \text{rad}(\widetilde{z}) = 0$, the bound is linearly decreasing for increasing radius r as long as no additional terms $\underline{x}_j \mathbf{d}_j^+$ or $\overline{x}_j \mathbf{d}_j^-$ occur in the two sums. For large values $|\underline{x}_j|$ and $|\overline{x}_j|$ the bound may be a poor, if one of the products $\underline{x}_j \mathbf{d}_j^+$ or $\overline{x}_j \mathbf{d}_j^-$ appear in the sums. In this case we recommend to put these indices into J^∞. Then the few equations corresponding to these indices must be added, but the bound will be better.

The additional computational costs for computing $\underline{f}^*$ are of order $0((m+p)\cdot n)$. The computed solution $(\widetilde{y}, \widetilde{z}, \widetilde{u}, \widetilde{v})$ approximates an optimal vertex of the dual problem yielding at most n nonzero components. Hence, at most n components of $\mathbf{y}$, and $\mathbf{z}$ are nonzero, and the lower bound needs only $O(n^2)$ operations.

The approximate solution can be computed by any lp-method for large or sparse systems. Since the lower bound requires only few matrix-vector operations, the sparse structure can be easily utilized. Moreover, degeneracy of the lp-problem causes not any difficulties.

Example. We consider the dual problem of (28):

$$\max 2y_1 + 2y_2 + 2v_1 + 2v_2 + 2v_3 \text{ subject to}$$

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 2 & 2 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} + \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} + \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \\ -4 \end{pmatrix} \tag{46}$$

$$y_1, y_2, v_1, v_2, v_3 \leq 0, \quad u_1, u_2, u_3 \geq 0$$

A short inspection shows that a dual optimal solution is given by $y_1^* = y_2^* = -1$ and all other variables are set equal zero. The optimal value is $g^* = -4 = f^*$. Also the dual problem is degenerate, since less than three components of the optimal solution are nonzero, and the dual optimal solution is not unique. For example the point $y_1^* = -\frac{1}{2}, y_2^* = -\frac{1}{2}, v_1{}^* = -\frac{1}{2}, v_2{}^* = -\frac{1}{2}$ is also optimal.

To obtain a lower bound $\underline{f}^*$, we set

$$\mathbf{y} := \begin{pmatrix} y_1^* \\ y_2^* \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \end{pmatrix} \leq 0,$$

and a short computation for the interval problem as defined in the previous section yields

$$\mathbf{d}_1 = [-2r, 2r], \quad \mathbf{d}_2 = [-2r, 2r], \quad \mathbf{d}_3 = [-3r, 3r],$$

and the lower bound

$$\begin{aligned} \underline{f}^* &= [2-r, 2+r]\mathbf{y}_1 + [2-r, 2+r]\mathbf{y}_2 \\ &\quad +2[-2r, 0] + 2[-2r, 0] + 2[-3r, 0] \\ &= -4 - 16r. \end{aligned} \tag{47}$$

Example. In order to illustrate some other effects, we consider the lp-problem

$$\min c^T x \quad \text{subject to} \quad a_{11}x \le a_1, a_{21}x \le a_2, \underline{x} \le x \le \overline{x}, \tag{48}$$

with integer coefficients and simple bounds

$$c = 100,\ a_{11} = 100,\ a_1 = 102,\ a_{21} = -90,\ a_2 = -90,\ \underline{x} = -2,\ \overline{x} = 2. \tag{49}$$

Because of the two inequalities $100x \le 102$, $-90x \le -90$ the set of feasible solutions satisfies $1 \le x \le 1.02$. Hence, this is a bounded and well-conditioned problem with optimal solution $x^* = 1$ and optimal value $f^* = 100$. Now, we allow that all coefficients a of this problem (with exception of the simple bounds) can vary within a radius r, that is we consider intervals of the form $[a - r, a + r]$.

For $r \ge 1$ this is a slightly complicated onedimensional interval problem. The difficulties arise, because some real problems within the interval input data contain no feasible solutions, some contain feasible solutions, and some are degenerate and ill-posed. Choosing

$$c = 100,\ a_{11} = 101,\ a_1 = 101,\ a_{21} = -89,\ a_2 = -91, \tag{50}$$

yields coefficients within the radius $r = 1$, and the inequalities $101x \le 101$, $-89x \le -91$ are contradictory. Hence, this problem is not feasible. If we change two coefficients within the interval input data to

$$a_{21} = -90,\ a_2 = -90, \tag{51}$$

then we obtain an ill-posed degenerate problem which has exactly one feasible solution $x = 1$.

The dual problem of (49) is

$$\begin{aligned} &\max 102y_1 - 90y_2 - 2u + 2v \quad \text{subject to} \\ &100y_1 - 90y_2 + u + v = 100,\ y_1, y_2, v \le 0,\ u \ge 0, \end{aligned} \tag{52}$$

yielding the dual optimal solution

$$y_1^* = u^* = v^* = 0,\ y_2^* = -\frac{10}{9}. \tag{53}$$

Assume that the approximate solution coincides with the dual optimal solution, then step(2) of our algorithm yields

$$\begin{aligned} &\mathbf{y}_1 = y_1^* = 0, \mathbf{y}_2 = y_2^* = -\frac{10}{9}, \quad \text{and} \\ &\mathbf{d} = [100 - r, 100 + r] - [-90 - r, -90 + r] \cdot (-\frac{10}{9}) = [-\frac{19r}{9}, \frac{19r}{9}]. \end{aligned} \tag{54}$$

Now with formula (36) we obtain the lower bound

$$\inf_{P \in [P]} f^*(P) \ge \min([-90 - r, -90 + r] \cdot (-\frac{10}{9}) - 2\frac{19r}{9} + 2(-\frac{19r}{9})) = 100 - \frac{86r}{9}. \tag{55}$$

The lower bound decreases linearly for increasing radius r, and coincides with the optimal value in the case $r = 0$. This is consistently with Theorem 6.1. Moreover, the influence of the simple bounds can be seen. In the example case of $\underline{x} := 0$, $\overline{x} := 2$ the lower bound is $100 - 48r/9$, demonstrating that some constraint propagation heuristic for improving the simple bounds can be very useful.

We remark that none of the $O(n^3)$ methods mentioned in Section 1 can compute any bounds for this example, since these methods additionally verify that all real problems within the interval problem have optimal solutions and are well-posed, which is not true for this example.

The previous example starts with a midpoint problem which is well-posed and has a finite optimal value. The assumptions of Theorem 6.1 do not require that even one real problem $P \in \mathbf{P}$ has primal feasible solutions. It is interresting to see what happens with the lower bound (36) in such situations.

Example. We consider again example (48) with input data (50). This problem has an empty set of feasible solutions, and $f^* = +\infty$. If we apply *linprog* to this problem, then we get the correct message

the primal appears to be infeasible (and the dual unbounded).

and the approximate values[2]

$$\widetilde{f} = 1.015200942446940 \times 10^2, \quad \widetilde{x} = 1.015200942446940$$

for the primal problem, and

$$\widetilde{y}_1 = -7.846817927076620 \times 10^7, \quad \widetilde{y}_2 = -8.904815962188074 \times 10^7,$$
$$\widetilde{u} = 2.039165444641289 \times 10^{-10}, \quad \widetilde{\widetilde{v}} = -7.434652599734950 \times 10^{-9}$$

for the dual problem. Obviously, the calculated approximations $\widetilde{f}, \widetilde{x}$ are completely wrong. But surprisingly, the approximations of the dual problem satisfy the dual constraint very sharp, and consequently we obtain the lower bound $1.780964192437591 \times 10^8$ which is clearly nearer to $f^* = +\infty$ than $\widetilde{f}$. In a branch and bound framework such a bound can serve to eliminate a subproblem. For varying radius of the interval input data the lower bound (36) is of the same order of magnitude for all radii $0 \le r \le 0.21$. But for $r = 0.22$ we get $-6.171554418163598 \times 10^6$, the sign has changed, and this bound is poor. The reason is that for values $r \le 0.21$ the first two terms in (36) remain much larger than the last two sums. This behaviour changes for values close to or greater than $r = 0.22$.

In the following we consider the algorithm for computing the rigorous lower bound in the general case. This algorithm is very similar to the algorithm for computing an upper bound. Condition (ii) of Theorem 6.1 requires to solve a linear interval system, which can be carried out as before.

Additionally, variables x_j with $j \in J^r$ may occur which are bounded only on one side. Then the inequalities (34) and (35) must be satisfied. In order to obtain appropriate interval vectors $\mathbf{y}$ and $\mathbf{z}$, we solve a slightly pertubated linear programming problem. If we take the midpoint problem $\check{\mathbf{P}} := (\check{\mathbf{A}}, \check{\mathbf{B}}, \check{\mathbf{a}}, \check{\mathbf{b}}, \check{\mathbf{c}})$ and change $\check{\mathbf{c}}$ to $c(\varepsilon)$ componentwise by

$$c_j(\varepsilon) := \begin{cases} \check{\mathbf{c}}_j + \varepsilon_j, & \text{if } \underline{x}_j = -\infty \\ \check{\mathbf{c}}_j - \varepsilon_j, & \text{if } \overline{x}_j = +\infty \\ \check{\mathbf{c}}_j & \text{otherwise,} \end{cases} \tag{56}$$

where $\varepsilon_j > 0$, then the exact dual optimal solution $(y(\varepsilon)^*, z(\varepsilon)^*, u(\varepsilon)^*, v(\varepsilon)^*)$ of this pertubated problem $P(\varepsilon)$ (provided the solution exists) satisfies for $\underline{x}_j = -\infty$ the equation

$$\check{\mathbf{c}}_j + \varepsilon_j = (\check{\mathbf{A}}_{:j})^T y(\varepsilon)^* + (\check{\mathbf{B}}_{:j})^T z(\varepsilon)^* + v_j(\varepsilon)^*. \tag{57}$$

Notice, that $u_j{}^*(\varepsilon)$ must be zero because of the term $\underline{x}_j u_j$ in the objective function of the dual problem. Hence,

$$d_j(\varepsilon) := \check{\mathbf{c}}_j - (\check{\mathbf{A}}_{:j})^T y(\varepsilon)^* + (\check{\mathbf{B}}_{:j})^T z(\varepsilon)^* = v_j(\varepsilon)^* - \varepsilon_j < 0, \tag{58}$$

and similarly for $\overline{x}_j = +\infty$ we obtain

$$d_j(\varepsilon) := \check{\mathbf{c}}_j - (\check{\mathbf{A}}_{:j})^T y(\varepsilon)^* + (\check{\mathbf{B}}_{:j})^T z(\varepsilon)^* = u_j(\varepsilon)^* - \varepsilon_j < 0, \tag{59}$$

Summarizing, for the real perturbed problem $P(\varepsilon)$ the conditions (34) and (35) are fulfilled, provided $\varepsilon_j > 0$ and a sufficiently good approximation $\widetilde{y}, \widetilde{z}, \widetilde{u}, \widetilde{v}$ of the optimal solution of $P(\varepsilon)$ is known. In order that these conditions also hold true for the interval problem $\mathbf{P}$, we have to comprise the interval input data and the rounding errors into ε_j as for the upper bound in the previous section.

We assume that approximate dual solutions $\widetilde{y}, \widetilde{z}$ of the midpoint problem $\check{\mathbf{P}}$ are known. Nothing is assumed about the quality of these approximations. The algorithm consists of the following steps:

[2] In this example we display all decimal digits because cancellation occurs in the dual constraint.

(1) Compute the deflation parameters

$$\begin{aligned}\varepsilon_j &= \alpha(\text{rad}(\mathbf{c}_j) + \text{rad}(\mathbf{A}_{:j})^T|\widetilde{y}| + \text{rad}(\mathbf{B}_{:j})|\widetilde{z}|) \\ &\quad +\alpha \cdot \varepsilon(|c_j| + |(A_{:j})^T|\,|\widetilde{y}| + |(B_{:j})^T|\,|\widetilde{z}|) + \eta,\end{aligned} \tag{60}$$

for $j \in J^r$.

(2) Define the perturbated problem $P(\varepsilon) := (\check{\mathbf{A}}, \check{\mathbf{B}}, \check{\mathbf{a}}, \check{\mathbf{b}}, c(\varepsilon))$ where $c(\varepsilon)$ is given by (56).

(3) Compute an approximate dual solution $\widetilde{\widetilde{y}}, \widetilde{\widetilde{z}}$ of the perturbated problem $P(\varepsilon)$ using the starting point $\widetilde{y}$, $\widetilde{z}$.
If the lp-solver cannot find an approximate solution, then STOP: *No lower bound for the optimal value.*

(4) Redefine $\widetilde{\widetilde{y}}$ by setting all positive components equal to zero, and set $\widetilde{y} := \widetilde{\widetilde{y}}$, $\widetilde{z} := \widetilde{\widetilde{z}}$.

(5) If $|J^\infty| = 0$ set $\mathbf{y} := \widetilde{y}$ and $\mathbf{z} := \widetilde{z}$. Otherwise, choose an index set $I = \{\beta_1, \cdots, \beta_{|J^\infty|}\}$ consisting initially of the indices corresponding to the components of $\widetilde{z}$, and if more indices for I are required ($i.e. |J^\infty| > p$), then choose those following indices which correspond to the components of y with largest absolute values. For $\widetilde{y}$, $\widetilde{z}$ and I compute with the algorithm for nonsquare linear interval systems (cf. Section 4) interval vectors $\mathbf{y}$ and $\mathbf{z}$ such that condition(ii) of Theorem 6.1 is satisfied. If this algorithm does not compute enclosures $\mathbf{y}$ and $\mathbf{z}$, then STOP: *No lower bound for the optimal value.*

(6) If the conditions (i) and (iii) of Theorem 6.1 are satisfied, then STOP: *lower bound is* $\underline{f}^*$.

(7) Set $\varepsilon_j = \alpha(\varepsilon_j + \eta)$ for $j \in J_r$.

(8) If the lp-solver gives no warning then goto step (2).

The linear interval system (32) must be solved with $m + p$ variables and $|J^\infty|$ equations. Notice, that in many applications $m+p$ is much larger than the number of unbounded variables $|J^\infty|$, yielding only a small quadratical linear interval system.

Several modifications of this algorithms are possible and may yield improvements at least for special classes of problems. For example, in cases where some of the simple bounds are zero and some are very large, it is advantegeous to choose the pertubated problem by defining

$$c_j(\varepsilon) := \begin{cases} \check{\mathbf{c}}_j + \varepsilon_j, & \text{if } |\underline{x}_j| \geq |\overline{x}_j| \\ \check{\mathbf{c}}_j - \varepsilon_j, & \text{if } |\underline{x}_j| < |\overline{x}_j| \end{cases} \tag{61}$$

Also other choices of the deflation parameters may improve the computing time and the bounds.

7 Applications

A frequently used approach for solving global optimization problems is to use a branch and bound framework together with relaxations; see for example Floudas [4] and Quesada and Grossmann [17], and the literature cited over there. A linear relaxation is a lp-problem such that each feasible solution of the global optimization problem is also feasible for the linear relaxation, and the linear objective function is an underestimator of the original objective function. Hence, a linear relaxation provides a lower bound for the global minimum value.

In order to discard subproblems in a branch and bound algorithm, it is important to have fast methods available for computing rigorous lower bounds for the optimal value of a linear relaxations. If this lower bound is greater than a known upper bound for the global minimum value, then this subproblem can be eliminated, because it cannot contain a global minimizer.

Therefore, computing rigorous upper bounds of the optimal value and the certificate that optimal solutions exist is also important. But in contrast to the rigorous lower bound, which has to be computed in each branching step, the computation of the rigorous upper bound and the certificate of existence of optimal solutions is necessary once only for the best approximate solution at the end of the branch and bound algorithm. However, it is important that also the upper bound should be computable in the degenerate case and for large systems.

A similar situation reproduces for mixed integer problems. There the linear relaxation is defined by relaxing the integer variables, i.e. the integer variables are treated as continuous variables. Rigorous bounds of linear programming problems are also required for verification methods of nonlinear minimax problems. For details the reader is referred to the recent preprint of Kearfott [8].

8 Numerical Experiments

The following results were obtained by using MATLAB [10] and the interval toolbox INTLAB (see Rump [21]). In all experiments interval input data $\mathbf{P}$ are considered with $\mathrm{rad}(\mathbf{P}) = r \cdot \check{\mathbf{P}}$, that is relative pertubations with respect to the midpoint problem are treated.

At first, we look at the two examples of Section 2. For the first example, defined by (1) and (2), we obtain for the simple upper bounds $\overline{x} = (30, 30, 30)^T$ and $r = 1.0 \times 10^{-5}$ the rigorous upper bound $\overline{f}^* = -5.99982 \times 10^1$ and the rigorous lower bound $\underline{f}^* = -6.00042 \times 10^1$. Since both bounds are finite, Theorem 5.1 shows that all lp-problems within the interval data have optimal solutions, and moreover the computed enclosure

$$\mathbf{x} = \begin{pmatrix} [2.421003440006751 \times 10^{-14}, 2.421003440006753 \times 10^{-14}] \\ \\ [7.477872102096182 \times 10^{-16}, 7.477872102096187 \times 10^{-16}] \\ \\ [9.999799997799970, 1.000020000219998] \end{pmatrix}$$

contains for every lp-problem a primal feasible solution being close to optimality. For $r = 1.0 \times 10^{-2}$ the lower bound $\underline{f}^* = -6.4200 \times 10^1$ and the upper bound $\overline{f}^* = -5.8174 \times 10^1$ are computed. In the case of simple upper bounds $\overline{x} = (10^{10}, 10^{10}, 10^{10})^T$ and r as before, although *linprog* gives the warning that *both the primal and the dual appears to be infeasible*, we obtain somewhat surprisingly $\underline{f}^* = -2.42318 \times 10^3$ and $\overline{f}^* = -4.88103 \times 10^1$, and thus the certificate that for all lp-problems within the interval data there exist optimal solutions. The computed enclosure is

$$\mathbf{x} = \begin{pmatrix} [4.631674545609419 \times 10^{-3}, 4.631674545609423 \times 10^{-3}] \\ \\ [5.591684607388289, 5.591684607388292] \\ \\ [4.403483715866093, 4.403883720266105] \end{pmatrix}.$$

Typically, in branch and bound algorithms a subproblem is discarded if the lp-solver detects infeasibility (see for example Borchers and Mitchell [3]). Observe that the previous lower bound, although very poor compared with the optimal value, would prevent a branch and bound algorithm from discarding subproblems which contain optimal points.

For the second example (cf. (3)) we display in Table 8.1 below for the dimensions $n = 8, \cdots, 12$ and fixed radius $r = 1.0 \times 10^{-12}$ the 2-norm condition number of A, the symbol $*$ if *linprog* gives the warning that *the primal appears to be infeasible (and the dual unbounded)*, and the error bounds $\underline{f}^*$ and $\overline{f}^*$.

Until dimension $n = 11$ rigorous upper and lower bounds are computed, implying the existence of optimal solutions for all lp-problems within the interval input data, and furthermore the upper bound is close to the optimal value. The lower bound is poor for $n \geq 10$, which is caused by the unsatisfactory dual approximation computed by *linprog*. However, the primal approximation and the enclosure $\mathbf{x}$ are rather fair regarding the large condition. Below we display the components $\mathbf{x}(1), \mathbf{x}(11), \mathbf{s}(1)$, and $\mathbf{s}(11)$ in the case of dimension $n = 11$:

$$\begin{pmatrix} [0.9999986005449631, 1.000001399410638] \\ \\ [0.9902735279885911, 1.009726272224163] \\ \\ [1.779069799688461 \times 10^{-15}, 1.779069799688465 \times 10^{-15}] \\ \\ [1.623200692326961 \times 10^{-21}, 1.623200692326963 \times 10^{-21}] \end{pmatrix}$$

Note that the optimal solution for the midpoint problem ${x_i}^* = 1, {s_i}^* = 0$ for $i = 1, \cdots, n$ is close to this enclosure. Only for dimension $n \geq 12$ an upper bound cannot be computed, and the existence of optimal solutions cannot be proved. This is not surprising since the radius is almost equal to the reciprocal of the condition number.

n	Condition	Warning	$\underline{f}^*$	$\overline{f}^*$
8	2.0×10^7		-3.3600×10^{-18}	2.1816×10^{-28}
9	2.9×10^8	*	-9.0261×10^{-4}	9.2276×10^{-20}
10	4.5×10^9	*	-3.6902×10^7	4.1743×10^{-13}
11	6.0×10^{10}	*	-1.1294×10^8	3.3209×10^{-15}
12	8.7×10^{11}	*	-3.4427×10^8	$+\infty$

Table 8.1 *Results for the second example (3)*

Last, we consider some random test problems with regard to the construction in Rosen and Suzuki [18]. All components of the primal solution x^* are uniformly distributed in the interval $(0, 1)$. The lower and upper bounds $\underline{x}_i$ and $\overline{x}_i$ are set equal to zero and one for every component x_i, respectively. The dual vector y^* is generated by uniformly distributing $n - p$ components in $(-1, 0)$ while the remaining components are set equal to zero. The coefficients of z^*, A and B are uniformly distributed in $(-1, 1)$. The right hand side a is chosen such that the first $n - p$ inequalities are active. The right hand side $b := Bx^*$. The coefficients of the objective function are generated by the equation $c := A^T y^* + B^T z^*$. This construction ensures that the optimum is known; that is, x^* is primal optimal and $(y^*, z^*, 0, 0)$ is dual optimal. For the following numerical results we consider interval input data with $\text{rad}(\mathbf{P}) = r \cdot \check{\mathbf{P}}$, $r := 10^{-8}$, and $\check{\mathbf{P}}$ is the previously defined random problem.

In Table 8.2 we display the number of inequalities m, the number of variables n, the number of equations p, the relative error $|\overline{f}^* - \underline{f}^*|/|\underline{f}^*|$ of the rigorous bounds, the time in seconds t_s reqired by *linprog*, the time

$t_{\underline{f}^*}$ for the lower bound, and the time $t_{\overline{f}^*}$ required by the upper bound.

m, n, p	$\lvert\overline{f}^* - \underline{f}^*\rvert / \lvert\underline{f}^*\rvert$	t_s	$t_{\underline{f}^*}$	$t_{\overline{f}^*}$
20, 10, 0	1.5777×10^{-6}	3.7×10^{-1}	3.6×10^{-2}	2.2×10^{0}
100, 20, 0	7.3356×10^{-7}	2.5×10^{0}	8.3×10^{-2}	3.1×10^{0}
300, 30, 0	2.8784×10^{-6}	3.9×10^{1}	1.3×10^{-1}	4.3×10^{1}
500, 40, 0	2.7805×10^{-6}	1.9×10^{2}	2.4×10^{-1}	1.8×10^{2}
1000, 50, 0	$*^1 : \underline{f}^* = 1.5015 \times 10^{1}$ $\overline{f}^* = \infty$	3.2×10^{2}	6.3×10^{-1}	3.3×10^{2}
0, 20, 10	1.01822×10^{-5}	1.8×10^{-1}	3.4×10^{-2}	4.5×10^{-1}
0, 100, 30	$*^2 : 5.1683 \times 10^{-2}$	3.9×10^{-1}	5.9×10^{-2}	2.0×10^{0}
0, 1000, 300	$*^2 : 2.1452 \times 10^{-2}$	2.5×10^{2}	4.8×10^{0}	2.9×10^{2}
20, 10, 5	4.5946×10^{-5}	3.4×10^{-1}	6.0×10^{-2}	4.2×10^{0}
50, 20, 10	2.0092×10^{-6}	1.2×10^{0}	7.9×10^{-2}	4.9×10^{0}
200, 40, 20	$*^2 : 7.2827 \times 10^{-2}$	1.9×10^{1}	1.4×10^{-1}	1.2×10^{2}

Table 8.2 *Results for randomly generated problems.*

The symbol $*^1$ means that in this case *linprog* gave the warning

the dual appears to be infeasible (and the primal unbounded)

and has computed poor approximate optimal solutions with accuracy of only two decimal digits. Note that because of the above construction there exist optimal solutions. The symbol $*^2$ means that *linprog* gave no warning, but the accuracy of the dual optimal solution was poor, yielding a poor rigorous lower bound. However, the upper bound $\overline{f}^*$ was very accurate.

From Table 8.2 it follows that the bounds are close together with respect to the interval data, provided sufficiently well approximations are calculated by *linprog*. The time for the lower bound $t_{\underline{f}^*}$ is only a fraction of the time t_s needed by the lp-solver. The time $t_{\overline{f}^*}$ is larger or of same order of magnitude as t_s. The reason is that for the upper bound some calls of *linprog* are required. But this routine ignores starting points such that a complete new algorithm is started yielding more compuatinal work than necessary.

Because of the curious results, we discuss separately from the table above the randomly generated problem in the case where $m = 500$, $n = 100$, and $p = 50$. By our construction we know the optimal value $f^* = 2.6136 \times 10^1$. The routine *linprog* gave the warning that *the dual appears to be infeasible (and the primal unbounded)*, and has computed the poor approximate optimal value -1.5733 and an approximation of a primal feasible point which is not within the simple bounds. One consequence was that the rigorous upper bound is equal to $+\infty$. However, the rigorous lower bound was equal to 8.8983, more close to the optimal value than the approximate value. The required time for *linprog* was 2538 seconds, and the time for the lower bound was 0.7 seconds.

9 Conclusions and Future Work

In this paper algorithms for computing rigorous lower and upper bounds of the optimal value of a linear programming problem are considered. It turns out that such bounds can be useful, because even commercial solvers like the MATLAB routine *linprog* may compute curios and unsatisfactory results, even for well-posed problems.

Especially, if linear relaxations are used in branch and bound methods then these rigorous bounds are advantaguous, because the user cannot see and judge the intermediate results, and solutions of the original problem cannot drop away. A benefit is that the lower bound, which must be computed in each branching step, needs only a small split of the computing time required for the lp-solver.

For interval input data the bounds allow a rough sensitivity analysis. This is usually not possible in classical sensitivity analysis for linear programming problems where only special input data are permitted to vary.

In our future work we want to generalize these results to convex programming problems, and to use these bounds in algorithms for mixed nonlinear integer problems. Moreover, by using another software than the MATLAB optimization toolbox, we want to investigate experimentally the range of applicability of these rigorous error bounds for large and sparse problems.

References

[1] G. Alefeld and J. Herzberger. *Introduction to Interval Computations.* Academic Press, New York, 1983.

[2] H. Beeck. Linear Programming with Inexact Data. Technical Report 7830, Abteilung Mathematik, TU München, 1978.

[3] B. Borchers and J.E. Mitchell. An Improved Branch and Bound Algorithm for Mixed Integer Nonlinear Programs. *Computers Ops Res.*, 21(4):359–367, 1994.

[4] C.A. Floudas. *Nonlinear and Mixed-Integer Optimization.* Oxford University Press, New York, 1995.

[5] ILOG CPLEX 7.1, User's manual. *ILOG.* France, 2001.

[6] C. Jansson. A Self-Validating Method for Solving Linear Programming Problems with Interval Input Data. *Computing*, Suppl. 6:33–45, 1988.

[7] R.B. Kearfott. *Rigorous Global Search: Continuous Problems.* Kluwer Academic Publisher, Dordrecht, 1996.

[8] R.B. Kearfott. On Verification of Solutions to Nonlinear Minimax Problems. Manuscript, 2002.

[9] R. Krawczyk. Fehlerabschätzung bei linearer Optimierung. volume 29 of *Lecture Notes in Computer Science*, pages 215–222. Springer Verlag, Berlin, 1975.

[10] MATLAB User's Guide, Version 6. The MathWorks Inc., 2000.

[11] R.E. Moore. *Methods and Applications of Interval Analysis.* SIAM, Philadelphia, 1979.

[12] NETLIB Linear Programming Library. `http://www.netlib.org/lp`.

[13] A. Neumaier. *Interval Methods for Systems of Equations.* Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1990.

[14] A. Neumaier. *Introduction to Numerical Analysis.* Cambridge University Press, 2001.

[15] A. Neumaier and O. Shcherbina. Safe bounds in linear and mixed-integer programming. Submitted for publication.

[16] F. Ordóñez and R.M. Freund. Computational expericence and the explanatory value of condition numbers for linear optimization. Technical Report MIT Operations Research Center Working Paper OR361-02, MIT, 2002. submitted for publication to SIAM J. Optimization. `http://web.mit.edu/rfreund/www/CVfreund.pdf`.

[17] I. Quesada and I.E. Grossmann. A Global Optimization Algorithm for Linear Fractional and Bilinear Programs. *Journal of Global Optimization*, (6):39–76, 1995.

[18] J.B. Rossen and S. Suzuki. Construction of Nonlinear Programming Test Problems. *Communication of ACM*, 8:113, 1965.

[19] S.M. Rump. Solving Algebraic Problems with High Accuracy. Habilitationsschrift. In U.W. Kulisch and W.L. Miranker, editors, *A New Approach to Scientific Computation*, pages 51–120. Academic Press, New York, 1983.

[20] S.M. Rump. Validated Solution of Large Linear Systems. In R. Albrecht, G. Alefeld, and H.J. Stetter, editors, *Computing Supplementum*, volume 9, pages 191–212. Springer, 1993.

[21] S.M. Rump. INTLAB - Interval Laboratory. http://www.ti3.tu-harburg.de/rump/intlab/index.html, 1998.

[22] R.J. Vanderbei. *Linear Programming: Foundations and Extensions*. Kluwer Academic Publishers, 1996.