Rigorous bounds for ill-posed linear programming problems

Marko Lange

Institute for Reliable Computing, Hamburg University of Technology

Keywords: Ill-posedness; Linear programming; Verification; Facial reduction; NETLIB LP library

Motivation

In [8], Ordóñez and Freund have shown that 71% of the instances from the Netlib test suite [7] - a benchmark suite containing difficult but practically relevant linear programming problems - have infinite condition measure. Due to their practical background and possible inaccuracies in the input data, these problem instances are very interesting for the application of verification methods.

However, the solvability principle of verification methods states that verification methods solve well-posed problems [10]: it is typically not possible to compute rigorous inclusions for the optimal value of an ill-posed problem using floating-point arithmetic because even the slightest perturbation may change its feasibility status. Without incorporating additional knowledge about the ill-posed instances from the NETLIB test suite, verification tools are not able to compute verified bounds for these problems (cf. [5]).

We are proposing an error free preprocessing procedure to replace a given ill-posed linear programming problem with an equivalent well-posed problem. We demonstrate the applicability of our procedure by computing new verified bounds for a large number of ill-posed problem instances from the NETLIB linear programming library.

Preliminaries

We are concerned with linear programming problems of the form

$$\begin{aligned} \inf_{x_f, x_l} \quad & c_f^T x_f + c_l^T x_l \\ & A_f x_f + A_l x_l = b \\ & x_l \geq 0, \end{aligned} \tag{LP}$$

where $x_f \in \mathbb{R}^{n_f}, x_l \in \mathbb{R}^{n_l}$ are free and non-negative decision variables, respectively, $A_\diamond \in \mathbb{R}^{m \times n_\diamond}$, $b \in \mathbb{R}^m$ and $c_\diamond \in \mathbb{R}^{n_\diamond}$ for $\diamond \in \{f, l\}$.

In accordance with Renegar's definition in [9], a feasible instance of (LP) is ill-posed if infinitesimal small perturbations can render the problem infeasible. This is precisely the case if the sets $\{A_f x_f + A_l x_l \mid x_l \geq 0\}$ and $\{b\}$ are separable by a hyperplane, but not strictly so.

If $u \in \mathbb{R}^m$ is a normal vector to such a hyperplane, then

$$(A_f x_f + A_l x_l)^T u \leq b^T u \tag{1}$$

is satisfied for all $x_f \in \mathbb{R}^{n_f}, 0 \leq x_l \in \mathbb{R}^{n_l}$. It is straightforward to show that (1) is equivalent to the conditions

$$(A_f x_f)^T u = 0, \quad (A_l x_l)^T u \leq 0, \quad b^T u \geq 0.$$

Since the separation is not strict, there exist x_f and a non-negative vector x_l such that (1) is satisfied with equality and therefore $b^T u = 0$. Summarizing, one can prove the following crucial equivalence.

Proposition. *A feasible instance of* (LP) *is ill-posed if, and only if, the conditions*

$$A_f^T u = 0, \quad A_l^T u \leq 0, \quad b^T u = 0 \tag{2}$$

are satisfied for non-trivial vectors $u \in \mathbb{R}^m$.

Reduction procedure

The conditions in (2) lead to another linear programming problem which can be used to compute suitable vectors u. If $u = 0$ is the only feasible solution to this problem, then (LP) is well-posed. Otherwise we found an indicator for ill-posedness.

Moreover, (2) can be used not only to detect ill-posedness but also for its removal. For any u satisfying (2) and every feasible point (x_f, x_l) of (LP), we have

$$0 = \underbrace{(A_f x_f + A_l x_l - b)^T}_{\equiv 0} u = \underbrace{x_l^T}_{\geq 0} \underbrace{A_l^T u}_{\leq 0}$$

and therefore

$$\forall 1 \leq i \leq n_l: \quad (A_l^T u)_i \cdot (x_l)_i = 0. \qquad (3)$$

If the equality constraints of (LP) are linearly independent, by which $u \neq 0$ implies $[A_l,\, A_f,\, b]^T u \neq 0$, then

$$\exists i: \quad (A_l^T u)_i \neq 0,\ (x_l)_i = 0.$$

In this case, (LP) can be reduced to a smaller equivalent linear programming problem by eliminating the respective entries $(x_l)_i$. On the other hand, if there are linearly dependent equality constraints, we may remove them and again obtain a reduced equivalent problem. This can be repeated up to the point where we derive a well-posed problem.

Following, one may use some verification tool for linear programming problems to compute verified bounds also for the original (LP).

Verification

The reduction approach is not exactly a new concept. The set described by the conditions in (2) is strongly related to the set of dual recession directions [2, 3] and our reduction approach is a specific form of facial reduction [1, 2, 6].

The actual difficulty lies in the verification of the reduction procedure. How can we compute rigorously an interval U that contains an actual solution to (2) but not the trivial vector of all zeros?

After using an auxiliary linear programming problem to compute an approximate solution to (2) and deciding which entries of u are nonzero via a simple threshold approach, the task reduces to determining and verifying linearly dependent equality constraints. By exploiting a line-up of presuppositions that are satisfied for most instances from the NETLIB linear programming library [7], it is actual possible to reduce the respective set of equality constraints to a linearly independent basis in a rigorous manner.

To demonstrate the applicability of our approach, in Table 1, we present relative duality gaps for some ill-posed instances from the NETLIB library. We chose problems for which VSDP [4] fails to compute rigorous bounds without prior preprocessing (PP). The used solver is SDPT3 [11].

Problem	only SDPT3	VSDP with (PP)
25FV47	$1.50 \times 10^{+3}$	1.94×10^{-8}
CZPROB	1.67×10^{-1}	1.14×10^{-8}
MODSZK1	$7.67 \times 10^{+8}$	1.02×10^{-2}
SCFXM1	1.42×10^{-10}	3.69×10^{-9}
SHIP12S	$8.70 \times 10^{+1}$	9.48×10^{-10}

Table 1: Duality gaps for selected LPs

References

[1] J. M. Borwein and H. Wolkowicz. Facial reduction for a cone-convex programming problem. *J. Austral. Math. Soc. Ser. A*, 30(3):369–380, 1981.

[2] V. Y.-L. Cheung. *Preprocessing and Reduction for Semidefinite Programming via Facial Reduction: Theory and Practice.* PhD thesis, University of Waterloo, Waterloo, Ontario, Canada, 2013.

[3] V. Y.-L. Cheung and H. Wolkowicz. Sensitivity analysis of semidefinite programs

without strong duality. Technical report, 2014.

[4] V. Härter, C. Jansson, and M. Lange. VSDP: A Matlab toolbox for verified semidefinite-quadratic-linear programming. `http://www.ti3.tuhh.de/jansson/vsdp/`, 2012.

[5] C. Keil and C. Jansson. Computational experience with rigorous error bounds for the netlib linear programming library. *Reliab. Comput.*, 12(4):303–321, 2006.

[6] B. F. Lourenço, M. Muramatsu, and T. Tsuchiya. Solving SDP completely with an interior point oracle. *ArXiv e-prints*, 2015.

[7] NETLIB linear programming library. A collection of linear programming problems, `http://www.netlib.org/lp/`.

[8] F. Ordóñez and R. M. Freund. Computational experience and the explanatory value of condition measures for linear optimization. *SIAM J. Optim.*, 14(2):307–333, 2003.

[9] J. Renegar. Incorporating condition measures into the complexity theory of linear programming. *SIAM J. Optim.*, 5(3):506–524, 1995.

[10] S. M. Rump. Verification methods: Rigorous results using floating-point arithmetic. *Acta Numer.*, 19:287–449, 2010.

[11] K.-C. Toh, M. J. Todd, and R. H. Tutuncu. On the implementation and usage of SDPT3 – a Matlab software package for semidefinite-quadratic-linear programming, version 4.0. Technical report, 2006.